A Polynomial Kernel for Block Graph Deletion

Eun Jung Kim[1] and O-joung Kwon[2]

1 **LAMSADE, CNRS – Université Paris Dauphine, France**
eunjungkim78@gmail.com

2 **Institute for Computer Science and Control, Hungarian Academy of Sciences, Hungary***
ojoungkwon@gmail.com

Abstract

In the Block Graph Deletion problem, we are given a graph G on n vertices and a positive integer k, and the objective is to check whether it is possible to delete at most k vertices from G to make it a block graph, i.e., a graph in which each block is a clique. In this paper, we obtain a kernel with $\mathcal{O}(k^6)$ vertices for the Block Graph Deletion problem. This is a first step to investigate polynomial kernels for deletion problems into non-trivial classes of graphs of bounded rank-width, but unbounded tree-width. Our result also implies that Chordal Vertex Deletion admits a polynomial-size kernel on diamond-free graphs. For the kernelization and its analysis, we introduce the notion of 'complete degree' of a vertex. We believe that the underlying idea can be potentially applied to other problems. We also prove that the Block Graph Deletion problem can be solved in time $10^k \cdot n^{\mathcal{O}(1)}$.

1998 ACM Subject Classification G.2.1 Combinatorics G.2.2 Graph Theory

Keywords and phrases block graph, polynomial kernel, single-exponential FPT algorithm

Digital Object Identifier 10.4230/LIPIcs.IPEC.2015.270

1 Introduction

In parameterized complexity, an instance of a parameterized problem consists in a pair (x, k), where k is a secondary measurement, called the *parameter*. A parameterized problem $Q \subseteq \Sigma^* \times N$ is *fixed-parameter tractable* (*FPT*) if there is an algorithm which decides whether (x, k) belongs to Q in time $f(k) \cdot |x|^{\mathcal{O}(1)}$ for some computable function f. Such an algorithm is called an *FPT* algorithm. We call an FPT algorithm a *single-exponential* FPT algorithm if it runs in time $c^k \cdot |x|^{\mathcal{O}(1)}$ for some constant c. A parameterized problem is said to admit a *polynomial kernel* if there is a polynomial time algorithm in $|x| + k$, called a *kernelization algorithm*, that reduces an input instance into an instance with size bounded by a polynomial function in k, while preserving the Yes/No answer.

Graph modification problems constitute a fundamental class of graph optimization problems. Typically, for a class Φ of graphs, a set Ψ of graph operations and a positive integer k, we want to know whether it is possible to transform an input graph into a graph in Φ by at most k operations chosen in Ψ. One of the most intensively studied graph modification problems is the Feedback Vertex Set problem. Given a graph G and an integer k as input, the Feedback Vertex Set problem asks whether G has a vertex subset of size at most k whose removal makes it a forest, which is a graph without cycles. The Feedback Vertex Set problem is known to admit an FPT algorithm [1, 11] and the

* Supported by ERC Starting Grant PARAMTIGHT (No. 280152).

© Eun Jung Kim and O-joung Kwon;
licensed under Creative Commons License CC-BY

10th International Symposium on Parameterized and Exact Computation (IPEC 2015).
Editors: Thore Husfeldt and Iyad Kanj; pp. 270–281
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

running time has been subsequently improved by a series of papers [23, 16, 14, 10, 4, 2, 7, 19]. Also, Thomassé [26] showed that it admits a kernel on $O(k^2)$ vertices.

The Feedback Vertex Set problem has been generalized to deletion problems for more general graph classes. *Tree-width* [25] is one of the basic parameters in graph algorithms and plays an important role in structural graph theory. Since forests are exactly the graphs of tree-width at most 1, the natural question is to decide, for an integer $w \geqslant 2$, whether there is an FPT algorithm with parameter k to find a vertex subset of size at most k whose removal makes it a graph of tree-width at most w (called Tree-width w Vertex Deletion). Courcelle's meta theorem [5] implies that the Tree-width w Vertex Deletion is FPT. Recently it is proved to admit a single-exponential FPT algorithm and a (non-uniform) polynomial kernel (a kernel of size $\mathcal{O}(k^{g(w)})$ for some function g) [12, 18].

On the other hand, there are interesting open questions related to two natural graph classes having tree-like structures. A graph is *chordal* if it does not contain any induced cycle of length at least 4. Chordal graphs are close to forests as a forest is a chordal graph without triangles. Marx [20] firstly showed that the Chordal Vertex Deletion problem is FPT, and Cao and Marx [3] improved that it can be solved in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$. However, it remains open whether there is a single-exponential FPT algorithm or a polynomial kernel [20, 3]. Another interesting class is *distance-hereditary graphs*, also known as graphs of *rank-width* at most 1 [22]. As many problems are tractable on graphs of bounded rank-width by the meta-theorem on graphs of bounded rank-width (equivalently, bounded clique-width) [6], it is worth studying the general Rank-width w Vertex Deletion problem. Again, it is known to be FPT from the meta-theorem on graphs of bounded rank-width [6], but for our knowledge, it is open whether there is a single exponential FPT algorithm or a polynomial kernel for this problem even for $w = 1$.

Block graphs lie in the intersection of chordal graphs and distance-hereditary graphs, and they contain all forests. A graph is a *block graph* if each block (maximally 2-connected subgraph) of it forms a clique. It is not difficult to see that block graphs are exactly those not containing an induced cycle of length at least 4 and a diamond (i.e. a cycle of length 4 with a single chord) as an induced subgraph. We study the following parameterized problem.

Block Graph Deletion
Input: A graph G, an integer k
Parameter: k
Question: Is there a vertex subset S of G with $|S| \leqslant k$ such that $G - S$ is a block graph?

Our main results are stated in the next two theorems.

▶ **Theorem 1.1.** *The* Block Graph Deletion *admits a kernel with $O(k^6)$ vertices.*

▶ **Theorem 1.2.** *The* Block Graph Deletion *can be solved in time $10^k \cdot n^{\mathcal{O}(1)}$.*

Our kernelization is motivated by the quadratic vertex-kernel by Thomassé [26]. In [26], basic reduction rules are applied so that whenever the size of the instance is still large, there must be a vertex of large degree (otherwise, it is a No-instance). Then a vertex v of large degree witnesses either so-called the *sunflower* structure, or the *2-expansion* structure. Our kernelization employs a similar strategy. In order to work with block graphs instead of forests, we come up with the notion of the *complete degree* of a vertex, which replaces the role of the usual degree of a vertex in Feedback Vertex Set. Also, we need to bound the size of a block which might appear in a block graph $G - S$, if such a set S of size at most k exists. Our single-exponential algorithm is surprisingly analogous to the algorithm of Chen. *et al.* [4] for Feedback Vertex Set although the analysis is non-trivial.

Since block graphs are exactly diamond-free chordal graphs, we have the following as a corollary of Theorem 1.1 and Theorem 1.2.

▶ **Corollary 1.3.** *On diamond-free graphs,* Chordal Vertex Deletion *admits a kernel with $O(k^6)$ vertices and can be solved in time $O(10^k \cdot n^{\mathcal{O}(1)})$.*

2 Preliminaries

All graphs considered in this paper are undirected and simple (without loops and parallel edges). For a graph G, we denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively. When we analyze the running time of an algorithm, we agree that $n = |V(G)|$. A *block tree* $\mathcal{T}_G$ of a graph G is the graph having $\mathcal{B} \cup \mathcal{C}$ as the vertex set, where $\mathcal{B}$ is the set of all blocks of G and $\mathcal{C}$ is the set of all cut vertices of G, and there is an edge $Bc \in E(\mathcal{T}_G)$ between $B \in \mathcal{B}$ and $c \in \mathcal{C}$ if and only if the cut vertex c belongs to the block B in G. The constructed graph does not contain a cycle. We say that a graph is a *block graph obstruction*, or simply an *obstruction*, if it is isomorphic to a diamond, or an induced cycle C_ℓ of length ℓ for some $\ell \geqslant 4$. A vertex is *simplicial* in G if $N_G(v)$ is a complete graph.

3 Complete degree of a vertex

We define a concept called the *complete degree* of a vertex in a graph. The definition of the complete degree is motivated by the following lemma, whose proof is deferred at the end of this section.

▶ **Proposition 3.1.** *Let G be a graph and let $v \in V(G)$ and let k be a positive integer. Then in $\mathcal{O}(kn^3)$ time, we can find either ?*
1. *$k+1$ obstructions that are pairwise vertex-disjoint, or*
2. *$k+1$ obstructions whose pairwise intersections are exactly the vertex v, or*
3. *$S_v \subseteq V(G)$ with $|S_v| \leqslant 7k$ such that $G - S_v$ has no block graph obstruction containing v.*

For a graph G and $v \in V(G)$ such that G has no $k+1$ vertex-disjoint obstructions and has no $k+1$ obstructions whose pairwise intersections are exactly the vertex v, the *complete degree* of v is defined as the minimum number of components of $G - (S_v \cup \{v\})$ among all possible $S_v \subseteq V(G) \backslash \{v\}$ where
- $|S_v| \leqslant 7k$, and
- $G - S_v$ has no block graph obstruction containing v.

Note that if $G - S_v$ has no block graph obstruction containing v, then $G[N_G(v) \backslash S_v]$ is a disjoint union of complete graphs.

To prove Proposition 3.1, we use the Gallai's A-path theorem. For a graph G and $A \subseteq V(G)$, an *A-path* of G is a path of length at least 1 whose end vertices are in A, and all internal vertices are in $V(G) \backslash A$.

▶ **Theorem 3.2** (Gallai [13]). *Let G be a graph and let $A \subseteq V(G)$ and let k be a positive integer. Then, in $\mathcal{O}(kn^2)$ time, we can find either ?*
1. *$k+1$ vertex-disjoint A-paths, or*
2. *$X \subseteq V(G)$ with $|X| \leqslant 2k$ such that $G - X$ has no A-paths.*

Proof of Proposition 3.1. Let $G_1 := (G - v) - E(G[N_G(v)])$. By Theorem 3.2, we can find in time $\mathcal{O}(kn^2)$ either
1. $2k+1$ vertex-disjoint $N_G(v)$-paths in G_1, or
2. $X \subseteq V(G)$ with $|X| \leqslant 4k$ such that $G_1 - X$ has no $N_G(v)$-paths.

Suppose that G_1 contains at least $2k+1$ pairwise vertex-disjoint $N_G(v)$-paths. Let P be one of these $N_G(v)$-paths in G_1 with p and q as its end vertices, and let P' be a shortest p,q-path in $G_1[V(P)]$. Note that P' has length at least 2. If P' has length 2, then $G[\{v\} \cup V(P')]$ is isomorphic to either C_4 or the diamond depending on the adjacency between p and q in G. If P' has length at least 3 and $pq \in E(G)$, then $G[V(P')]$ is an induced cycle of length at least 4. If P' has length at least 3 and $pq \notin E(G)$, then $G[\{v\} \cup V(P')]$ is an induced cycle of length at least 5. Thus, $G[\{v\} \cup V(P)]$ contains an obstruction, and G contains either disjoint $k+1$ obstructions, or $k+1$ obstructions whose pairwise intersections are exactly v.

So, we may assume that there exists $X \subseteq V(G_1)$ with $|X| \leqslant 4k$ such that $G_1 - X$ has no $N_G(v)$-paths. Now, we greedily find a maximal set $\mathcal{P}$ of vertex-disjoint induced P_3 in $G[N_G(v)]$ by searching vertex subsets of size 3. If there are $k+1$ vertex-disjoint induced P_3's, then G has $k+1$ diamonds whose pairwise intersections are exactly v. Otherwise, we set $S_v = X \cup \bigcup_{P \in \mathcal{P}} V(P)$ and notice that $|S_v| \leqslant 7k$. Observe that $G - S_v$ has no block graph obstruction containing v. Clearly, we can find $\mathcal{P}$ in time $\mathcal{O}(kn^3)$. ◀

In our algorithm, we need to find a vertex of sufficiently large complete degree and the corresponding deletion set S_v in polynomial time. However, we just need sufficiently many complete graphs on the neighborhood, and do not need to compute the complete degree of each vertex exactly. The following lemma will be used to analyze the difference between an optimal set and an arbitrary set S_v obtained by Proposition 3.1.

▶ **Lemma 3.3.** *Let G be a graph and let $S_1, S_2 \subseteq V(G)$ such that for each $1 \leqslant i \leqslant 2$, $G - S_i$ is a disjoint union of complete graphs. If $|S_2| \leqslant k$, then the number of components of $G - S_2$ is at least the number of components of $G - S_1$ minus k.*

4 Finding a vertex of large complete degree

In this section, we prove that if a graph is reduced under certain rules and its size is still large, then there should exist a vertex of large complete degree. To do this, we first provide basic reduction rules.

4.1 Basic reduction rules

▶ **Reduction Rule 1** (Block component rule). *If G has a component H that is a block graph, then we remove H from G.*

▶ **Reduction Rule 2** (Cut vertex rule). *Let v be a vertex of G such that $G - v$ contains a component H where $G[V(H) \cup \{v\}]$ is a connected block graph. Then we remove H from G.*

Two vertices v, w in a graph G are called *true twins* if $N_G(v) \backslash \{w\} = N_G(w) \backslash \{v\}$ and $vw \in E(G)$. Note that two simplicial vertices in a block of a block graph are true twins.

▶ **Reduction Rule 3** (Twin rule). *Let S be the set of vertices that are pairwise true twins in G. If $|S| \geqslant k+2$, then we remove vertices except $k+1$ vertices.*

It is not hard to observe that Rules 1, 2, and 3 are sound. Note that we can test whether a given graph is a block graph in quadratic time using an algorithm to partition the graph into blocks [15], and testing whether each block is a complete graph.

▶ **Reduction Rule 4** (Reducing block-cut vertex paths). *Let $t_1t_2t_3t_4$ be an induced path of G and for each $1 \leqslant i \leqslant 3$, let $S_i \subseteq V(G) \backslash \{t_1, \dots, t_4\}$ be a clique of G such that*

- *for each $1 \leqslant i \leqslant 3$ and $v \in S_i$, $N_G(v) \backslash S_i = \{t_i, t_{i+1}\}$, and*
- *for each $2 \leqslant i \leqslant 3$, $N_G(t_i) = \{t_{i-1}, t_{i+1}\} \cup S_{i-1} \cup S_i$.*

Then we remove S_2 and contract t_2t_3.

Clearly, we can apply Reduction Rule 4 in polynomial time. We prove the soundness of Reduction Rule 4 in the full version [17]. The following rule will be applied using Proposition 3.1.

▶ **Reduction Rule 5** ($(k+1)$-distinct obstructions rule). *Let $v \in V(G)$ and let $G' := G - v - E(G[N_G(v)])$ such that there are $2k+1$ vertex-disjoint $N_G(v)$-paths in G'. If G contains $k+1$ vertex-disjoint obstructions, then say that it is a* No*-instance. Otherwise, we remove v from G, and decrease k by one. (By Proposition 3.1, one of them exists.)*

4.2 A vertex of large complete degree

An instance (G, k) is called a *reduced instance* if it is reduced under Rules 1, 2, 3, 4, and 5 introduced in the previous subsection. In this subsection, we prove that there exists a vertex of large complete degree whenever a reduced instance is sufficiently large, which is stated as Theorem 4.1.

For positive integers k, ℓ, we define that

- $g_1(k, \ell) := 6k^2(\ell + 14k)^2 + 2k(\ell + 14k)$,
- $g_2(k, \ell) := (k+1)^2 + 7k^2 + \frac{1}{2}k(\ell + 14k)$.

▶ **Theorem 4.1.** *Let (G, k) be a reduced instance of* Block Graph Deletion *that is a* Yes*-instance. If G has at least $k + g_1(k, \ell)g_2(k, \ell)$ vertices then G has a vertex of complete degree at least $\ell + 1$.*

Let (G, k) be a reduced instance of Block Graph Deletion and let $S \subseteq V(G)$ of size at most k such that $G - S$ is a block graph. We let $G' := G - S$ and for each $v \in S$, we define that

- $G_v := G[V(G') \cup \{v\}]$,
- S'_v is a vertex set of size at most $7k$ in $G - v$ that is obtained by Proposition 3.1,
- $S_v := S'_v \cap V(G')$.

Let $T := \bigcup_{v \in S} S_v$. Note that $|T| \leqslant 7k^2$ and for each $v \in S$, there are no block graph obstructions containing v in $G_v - T$.

We first give a bound on the size of each block of G' and the number of blocks in G' sharing a cut vertex with it, assuming that there is no vertex in S of large complete degree in G. Each block of G' consists of the set of simplicial vertices and the set of cut vertices in G'.

▶ **Lemma 4.2.** *Let F be a graph whose vertex set is $X \cup \{v_1, \dots, v_t\}$ such that $t \geqslant 2$ and X is a clique of F and every two vertices of X have different neighbors on $\{v_1, \dots, v_t\}$. If $|X| \geqslant t + 2$, then F contains a diamond having exactly one vertex of $\{v_1, \dots, v_t\}$.*

Proof. Without loss of generality, we can assume that $\{v_1, \dots, v_t\}$ is a minimal set with the aforementioned property. Notice that there exists a vertex v_i which has at least two neighbors in X. By minimality assumption, v_i is not adjacent with all vertices in X. Choose distinct vertices $x, y, z \in X$ such that x,y are neighbors of v_i and z is not. Observe that $F[\{v_i, x, y, z\}]$ is isomorphic to the diamond containing exactly one vertex of $\{v_1, \dots, v_t\}$ ◀

▶ **Lemma 4.3.** *Let B be a block of G', and let B_1 and B_2 be the sets of all simplicial vertices and all cut vertices of G' contained in B, respectively. Let $H_1, H_2, \dots, H_t$ be the components of $G' - V(B)$ that has a neighbor in B. The followings hold.* ?

1. *$|B_1| \leqslant (k+1)^2 + 7k^2$.*
2. *If for every $v \in S$, v has complete degree at most ℓ in G, then $|B_2| \leqslant t \leqslant k(\ell + 14k)$.*

Proof. The proof for (2) can be found in the full version [17]. (1) We first give a bound on the number of simplicial vertices of a block for $G' - T$. Note that $B \backslash T$ is a block of $G' - T$. Let B_1' be $B_1 \backslash T$. Clearly, $|B_1| \leqslant |B_1'| + 7k^2$.

Since vertices in B_1' are pairwise true twins in G', if two vertices in B_1' have the same neighbors on S, then they are true twins in G. We partition B_1' into an equivalent classes where two vertices are equivalent if they have the same neighbors on S. From Reduction Rule 3, each equivalent class has at most $k + 1$ vertices.

If $|S| \leqslant 1$, then there are at most 2 equivalent classes in B_1'. If $|S| \geqslant 2$ and the number of equivalent classes in B_1' is at least $k + 2$, then since $|S| \leqslant k$, G contains a diamond containing exactly one vertex v of S by Lemma 4.2. This contradicts to the fact that $G_v - T$ has no obstruction containing v. Thus, the number of equivalent classes in B_1' is at most $k + 1$ and $|B_1| \leqslant |B_1'| + 7k^2 \leqslant (k + 1)^2 + 7k^2$. ◀

Contracted Block Tree. We introduce a notion called the *contracted block tree* of G. A contracted block tree $\mathcal{T}_G$ of a connected graph G is a rooted tree obtained from a block tree $\mathcal{T}_G^0$ of G by (i) choosing a block vertex of $\mathcal{T}_G^0$ as a root, and (ii) for each cut vertex c of $\mathcal{T}_G^0$, identifying it with its unique parent.

Let $\mathcal{T}_{G'}$ be the union of contracted block trees of connected components of G'. We color the vertices of $\mathcal{T}_{G'}$ in three phases: in the first phase, for every vertex $v \in S$ and for every $w \in N_{V(G) \backslash S}(v)$, we choose the (unique) block $B \in V(\mathcal{T}_{G'})$ which contains w and is closest to the root, and color B by red. Let R_1 be the vertices colored red so far. In the second phase, we again recursively color the least common ancestor of any pair of red vertices by red. Let R be the set of red vertices $\mathcal{T}_{G'}$. All other vertices of $\mathcal{T}_{G'}$ are colored blue.

▶ **Lemma 4.4.** *Suppose that the complete degree of v is at most ℓ for every $v \in S$. Then we have $|R| \leqslant 2k(\ell + 14k)$.*

▶ **Lemma 4.5.** *Let T be a tree with at least 2 vertices and degree at most d, and let M be a set of vertices in T. Then there are at most $d \cdot |M|$ connected components in $T - M$.*

The next lemma follows from Lemma 4.4 and 4.5.

▶ **Lemma 4.6.** *If G' contains at least $g_1(k, \ell)$ blocks, then $\mathcal{T}_{G'}$ has a blue component on at least 3 vertices.*

Proof of Theorem 4.1. Let (G, k) be a reduced instance with $|V(G)| \geqslant k + g_1(k, \ell)g_2(k, \ell)$ and $S \subseteq V(G)$ be a set of size at most k such that $G - S$ is a block graph. To derive contradiction, suppose that for every $v \in S$, v has complete degree at most ℓ in G. Then $G' = G - S$ has at least $g_1(k, \ell)g_2(k, \ell)$ vertices. Let p be the number of blocks of G'. From Lemma 4.3 and the fact that each cut vertex is contained in at least two blocks, we obtain $|V(G')| \leqslant p((k+1)^2 + 7k^2) + \frac{1}{2}pk(\ell + 14k) \leqslant p \cdot g_2(k, \ell)$. Therefore, we have $p \geqslant g_1(k, \ell)$. By Lemma 4.6, $\mathcal{T}_{G'}$ contains a blue component P on at least 3 vertices.

We claim that P is (i) a path, and (ii) each of its two end vertices, and no other, is adjacent with exactly one red vertex. Let us prove (i) first. Let W be the unique block vertex in P which is closest to the root. Notice that W is not the root itself since the instance is reduced with respect to Reduction Rule 1 and thus the root is a red vertex. Hence W has a unique parent which is red. For any Z which is a leaf in the subtree P, it is adjacent with at least one red vertex. Indeed, if not, Z is a leaf in $\mathcal{T}_{G'}$. Then by Reduction Rule 2, the block Z (possibly except for its unique cut vertex) should have been removed from G, a contradiction. Note that any red vertex adjacent with Z is a child of Z since the path from Z to W is blue and $W \neq Z$. Furthermore, the subtree P has exactly one leaf since

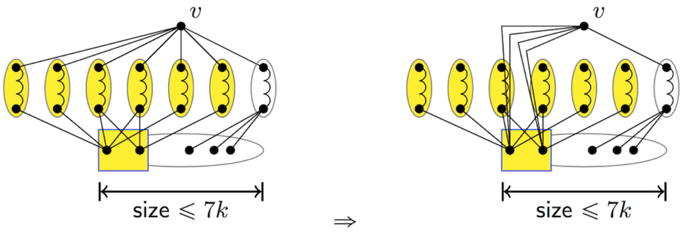

Figure 1 Reduction Rule 6.

otherwise, the second phase of coloring must have colored the branching vertices contained in P, a contradiction. This establishes (i). For (ii), observe that if (ii) does not hold, then some vertex of P must have been colored in the second phase, a contradiction.

Now, with P together with the two red vertices incident with $V(P)$, we can apply Reduction Rule 4, a contradiction. Therefore, we conclude that there exists a vertex $v \in S$ such that v has complete degree at least $\ell + 1$ in G. ◀

5 Reducing the instance with large complete degree

We introduce the last rule, which will be used when G has a vertex of large complete degree. We use the well-known technique, called the α-expansion lemma, which is already used in several kernelization algorithms [26, 9, 21, 8]. One notable difference from other approaches is that, to guarantee the equivalence, we add some paths in the given graph, and thus increase the number of vertices. However, we show that our rule decreases $n + m$ where m is the number of edges whose both degrees are at least 3, by using the 3-expansion le???mma instead of the 2-expansion lemma.

▶ **Reduction Rule 6** (Large complete degree rule). *Let $v \in V(G)$ and $X \subseteq V(G) \backslash \{v\}$ with $|X| \leqslant 7k$. Let $\mathcal{C}$ be a set of connected components of $G - (X \cup \{v\})$ and let $\phi : X \to \binom{\mathcal{C}}{3}$ such that*

- *for each $C \in \mathcal{C}$, $G[\{v\} \cup V(C)]$ is a block graph, v has a neighbor in C, and there exists a vertex $x \in X$ that has a neighbor in C,*
- *for $x \in X$, $\phi(x)$ is a subset of $\mathcal{C}$ where each graph in $\phi(x)$ has a neighbor of x, and*
- *the sets in $\{\phi(x) : x \in X\}$ are pairwise disjoint.*

Then, remove all edges between v and every component of $\mathcal{C}$, and add two internally vertex-disjoint paths of length two between v and each vertex $x \in X$. (All of the new vertices in these paths have degree 2 *in the resulting graph). If a component of $\mathcal{C}$ has a vertex of degree* 1 *in the resulting graph, then we remove the vertex. See Figure 1.*

We prove that Reduction Rule 6 is safe in the full version [17]. As we discussed, we clarify that it decreases $n + m^*$ where m^* is the number of edges whose both end vertices have degree at least 3. Since $|\mathcal{C}| \geqslant 3|X|$ and $n + m^*$ is increased by $2|X|$ by adding paths of length 2 from v to each vertex of X, it is sufficient to show that for each $C \in \mathcal{C}$, $n + m^*$ is decreased by at least 1 by removing the edges between v and C. Let $C \in \mathcal{C}$. If $|N_G(v) \cap C| \geqslant 3$, then it is trivial. First assume that $|N_G(v) \cap C| = 2$. Then C has more than two vertices, or there exists a vertex $x \in X$ that has a neighbor on $N_G(v) \cap C$. In either case, it is not difficult to verify that one of the vertex in $N_G(v) \cap C$ has degree at least 3 in G. Therefore, m^* is decreased by at least 1 when removing the edges between v and C. Now, let us assume that $N_G(v) \cap C = \{w\}$ for some $w \in V(C)$. If w has degree 2, then after removing the edge

vw, we also remove w following Reduction Rule 6. Thus, n is decreased by 1. Otherwise, removing vw decreases m^* by 1. We conclude that $n+m^*$ is always decreased when applying Reduction Rule 6.

Now we describe how to obtain a polynomial-size kernel from a given instance. The algorithm presented in the following theorem is used as a subroutine.

▶ **Theorem 5.1** (α-expansion lemma [26]). *Let α be a positive integer. Let F be a bipartite graph on the bipartition (X, Y) with $|Y| \geqslant \alpha|X|$ such that every vertex of Y has at least one neighbor in X. Then there exist nonempty subsets $X' \subseteq X$ and $Y' \subseteq Y$ and a function $\phi : X' \to \binom{Y'}{\alpha}$ such that*

- *$N_F(Y') \cap X = X'$,*
- *$\phi(x) \subseteq N_F(x)$ for each $x \in X'$, and*
- *the sets in $\{\phi(x) : x \in X'\}$ are pairwise disjoint.*

In addition, such pair of subsets X', Y' can be computed in polynomial time in $\alpha|V(F)|$.

Proof of Theorem 1.1. Given an instance (G, k), we exhaustively apply Reduction Rules 1-5 to obtain a reduced instance. If a reduced graph G has at least $k+g_1(k, 29k)g_2(k, 29k)$ vertices, then by Theorem 4.1, G has a vertex of complete degree at least $29k$. By Proposition 3.1, we can find in polynomial time a vertex v and a vertex set $S_v \subseteq V(G-v)$ such that $G - S_v$ has no block graph obstruction containing v, and $G[N_G(v)\backslash S_v]$ has at least $29k - 7k = 22k$ components. Note that there are at most k components of $G - (\{v\} \cup S_v)$ that may contain an obstruction, and for each component C of $G - (\{v\} \cup S_v)$, at most one components of $G[N_G(v)\backslash S_v]$ can be contained in C. Let $\mathcal{C}$ be the set of components of $G-(\{v\} \cup S_v)$ which (i) contains a component of $G[N_G(v)\backslash S_v]$, and (ii) has no block graph obstructions. Since $|\mathcal{C}| \geqslant 22k - k = 21k$ and $|S_v| \leqslant 7k$, using Theorem 5.1, we can find in polynomial time sets $\mathcal{C}' \subseteq \mathcal{C}$ and $S'_v \subseteq S_v$ and a function $\phi : S'_v \to \binom{\mathcal{C}'}{3}$ such that

- the set of vertices in S_v that has a neighbor in $\bigcup_{C \in \mathcal{C}'} V(C)$ is S'_v,
- for $x \in S'_v$, $\phi(x)$ is a subset of $\mathcal{C}$ where each graph in $\phi(x)$ has a neighbor of x, and
- the sets in $\{\bigcup_{C \in \phi(x)} V(C) : x \in S'_v\}$ are pairwise disjoint.

Note that for each $C \in \mathcal{C}'$, $G[\{v\} \cup V(C)]$ is a block graph, otherwise, it has an obstruction containing v, contradicting to the definition of S_v. Furthermore, for each $C \in \mathcal{C}'$, there exists a vertex $x \in S'_v$ that has a neighbor in C, otherwise, we can reduce it using Reduction Rule 2. So, we can apply Reduction Rule 6 to reduce this instance. We apply these reductions recursively. As we discussed, each step decreases $n + m^*$ where m^* is the number of edges whose both end vertices have degree 3, so, it will terminate in polynomial time, and at the final step, the resulting graph will have less than $k + g_1(k, 29k)g_2(k, 29k) = \mathcal{O}(k^6)$ vertices. ◀

6 A fixed parameter tractable algorithm

The goal of this section is to prove Theorem 1.2 claiming an $O(10^k \cdot n^{O(1)})$-time algorithm for Block Graph Deletion. We apply iterative compression technique, which is established as a powerful tool to design FPT algorithms since it was first introduced by Reed, Smith and Vetta [24]. Our algorithm Block Graph Deletion requires as a subroutine an FPT algorithm for the following disjoint version of Block Graph Deletion.

Disjoint Block Graph Deletion
Input: A graph G, $S \subseteq V(G)$ such that both $G - S$ and $G[S]$ are block graphs, an integer k.
Parameter: k
Task: Find a *solution* to (G, S, k), i.e. a set $\tilde{S} \subseteq V(G) \backslash S$ such that $G - \tilde{S}$ is a block graph and $|\tilde{S}| \leqslant k$, or correctly report that no such set exists.

We present an algorithm $\mathbf{Block}(G, S, k)$ which solves Disjoint Block Graph Deletion in time $O(3^{k+\ell} \cdot n^6)$, where ℓ is the number of connected components in $G[S]$.

Algorithm 1 Algorithm for Block Graph Deletion

```
 1: procedure Block($G, S, k$)
 2:     if $k \leqslant 0$ and $V(G)\backslash S \neq \varnothing$, return No.
 3:     if $k \geqslant 0$ and $G$ is a block graph, return $\varnothing$.
 4:     if $u, v, w \in V(G)\backslash S$ s.t. $G[S \cup \{u, v, w\}]$ is not a block graph then
 5:                           ▷ $u, v, w$ are not necessarily distinct if $|V(G)\backslash S| \leqslant 2$
 6:         Block($G - u, S, k - 1$) $\cup \{u\}$                    ▷ Small Set Branching Rule
 7:         Block($G - v, S, k - 1$) $\cup \{v\}$
 8:         Block($G - w, S, k - 1$) $\cup \{w\}$
 9:     else if there is $uv \in E(G - S)$ and $x, y \in N_S(\{u, v\})$ s.t.
10:         $x, y$ belong to distinct connected components of $G[S]$ then
11:         Block($G - u, S, k - 1$) $\cup \{u\}$                    ▷ Component Branching Rule
12:         Block($G - v, S, k - 1$) $\cup \{v\}$
13:         Block($G, S \cup \{u, v\}, k$)
14:     else
15:         Let $B$ be a leaf block of $G - S$ and $\partial_{G-S}(B) = \{b\}$.
16:         $G' \leftarrow G - B\backslash\partial_{G-S}(B) + \{bw : w \in N_S(B)\}$          ▷ Bypass Rule
17:         Block($G', S, k$).
18:     end if
19: end procedure
```

Let us establish that $\mathbf{Block}(G, S, k)$ correctly returns a solution to (G, S, k) if it is a Yes-instance, and returns No otherwise. Notice that if (G, S, k) does not meet the condition at line 3, then $V(G)\backslash S$ is non-empty and thus one of the steps at lines 2, 4, 10, or 15 will be executed and some output will be returned at the end of the algorithm $\mathbf{Block}(G, S, k)$. The execution of $\mathbf{Block}(G, S, k)$ can be represented by a search tree where each node corresponds to a call made during the execution. For the correctness of the algorithm, we use induction on the level of a call in the search tree. It is clear that lines 2–3, corresponding to the base case, returns the output correctly. If the condition at line 4 is met, then any solution $\tilde{S}$ to (G, S, k) must contain one of u, v and w. Conversely, if $\tilde{S}$ is a solution returned by one of the calls **Block** at lines 6–8, then $\tilde{S}$ together with u, v, or w is a solution to (G, S, k). To see the correctness of lines 11–13, first notice that they enumerate all possible intersection of a solution $\tilde{S} \cap \{u, v\}$. Hence it suffices to verify that $G[S \cup \{u, v\}]$ is indeed a block graph. This is a consequence from the fact that G does not meet the condition of line 4 for any (at most) three vertices.

The branching rules considered at lines 4-8 and lines 10-13 are called the `Small Set Branching` and `Component Branching`, respectively. Notice that an instance (G, S, k) considered at line 15 is *reduced with respect* to `Small Set Branching` and `Component Branching` or, simply put, *irreducible*: neither branching rules apply to (G, S, k). For the correctness of the algorithm **Block**, it remains to show that `Bypass Rule` at line 16 is safe, that is, $\tilde{S}$ is a solution to the instance (G', S, k) at line 17 if and only if it is a solution to (G, S, k). We need the following lemmata, whose proofs can be found in the full version [17].

▶ **Lemma 6.1.** *Let (G, S, k) be an irreducible instance and B be a leaf block of $G - S$. Then either $N_S(B) = \varnothing$ or there exists a single block X of $G[S]$ such that $N_S(B) \subseteq X$.*

▶ **Lemma 6.2.** *Let (G, S, k) be an irreducible instance and B be a leaf block of $G - S$. Then $G[S \cup B]$ is a block graph.*

▶ **Lemma 6.3.** *Let (G, S, k) be an irreducible instance and B be a leaf block of $G - S$. Then there exists a vertex $u \in B$ such that $N_S(u) = N_S(B)$.*

▶ **Lemma 6.4.** *Let (G, S, k) be an irreducible instance and B be a leaf block of $G - S$. If there is a vertex set $\tilde{S} \subseteq V(G) \backslash S$ such that $G - \tilde{S}$ is a block graph, then there is $\tilde{S}' \subseteq V(G) \backslash S$ such that $G - \tilde{S}'$ is a block graph, $|\tilde{S}'| \leqslant |\tilde{S}|$ and $\tilde{S}' \cap (B \backslash \partial_{G-S}(B)) = \varnothing$.*

The following lemma states the correctness of `Bypass Rule` applied at lines 15–17.

▶ **Lemma 6.5.** *Let (G, S, k) be an irreducible instance, B be a leaf block of $G - S$, and G' be the graph obtained by applying `Bypass Rule`. Then,*

- *if $\tilde{S}$ is a solution to (G, S, k), $\tilde{S} \backslash (B \backslash \partial_{G-S}(B))$ is a solution to (G', S, k), and*
- *if $\tilde{S}'$ is a solution to (G', S, k), it is also a solution to (G, S, k).*

Proof. Let b be the unique cut vertex of $G - S$ contained in B. Let us prove the first implication. Suppose that $\tilde{S}$ is a solution to (G, S, k) such that $\tilde{S} \cap (B \backslash \partial_{G-S}(B)) = \varnothing$. Such a solution exists by Lemma 6.4. We show that $\tilde{S}$ is a solution to (G', S, k), from which the first implication follows. If $b \in \tilde{S}$, then $G' - \tilde{S}$ clearly a block graph as it is an induced subgraph of $G - \tilde{S}$. Let us consider the case when $b \notin \tilde{S}$. For the sake of contradiction, suppose that $G' - \tilde{S}$ contains a vertex set C inducing an obstruction. Consider a vertex $u \in B$ such that $N_S(u) = N_S(B)$. The existence of such u is shown in Lemma 6.3. Note that $u \neq b$ and there exists $x \in N_S(B)$ such that $bx \notin E(G)$ and bx is contained in C, otherwise, C also appears in $G - \tilde{S}$. If C contains one more vertex from $N_S(B)$, then C should be a diamond with two intersections on $N_S(B)$ in $G' - \tilde{S}$. Then $G[V(C) \backslash \{b\} \cup \{u\}]$ is a diamond of $G - \tilde{S}$, which is a contradiction. Thus, $|V(C) \cap N_S(B)| = 1$ and $G[V(C) \cup \{u\}]$ induces a subgraph isomorphic to a graph obtained from C by subdividing one edge. It contains an obstruction in $G - \tilde{S}$, which contradicts to our assumption.

We establish the second implication. Suppose that $\tilde{S}'$ is a solution to (G', S, k), but $G - \tilde{S}'$ is not a block graph. Let C be a vertex set inducing an obstruction in $G - \tilde{S}'$. Then $G[C]$ is not a diamond nor a cycle of length 4 since otherwise, $G[C \cup S]$ is not a block graph and $|C \backslash S| \leqslant 3$, contradicting to the assumption that (G, S, k) is reduced with respect to `Small Set Branching`. Therefore $G[C]$ must be an induced cycle of length at least 5. Notice that C contains some vertex $v \notin B \cup S$ since $G[B \cup S]$ is a block graph by Lemma 6.2. There are two possibilities, and in each case we derive a contradiction.

When $b \notin C$: Notice that $N_S(B) \cap C$ is a separator between $B \cap C$ and v in $G[C]$, and thus contains a minimal separator between $B \cap C$ and v. However, $N_S(B)$ is a complete graph by Lemma 6.1 while any minimal separator in an induced cycle must be non-adjacent, a contradiction.

When $b \in C$: Observe that there is a vertex $x \in N_S(B) \cap C$ such that x is adjacent with some vertex, say w, in $B \cap C$. We claim that $N_S(B) \cap C = \{x\}$. Suppose not, and let y be a vertex in $(N_S(B) \cap C) \backslash \{x\}$. The existence of $v \in C \backslash (B \cup S)$ implies $wy \notin E(G)$. Take $u \in B$ such that $N_S(u) = N_S(B)$, which is possible due to Lemma 6.3, and observe that $ux, uy \in E(G)$. It follows that $G[\{u, w, x, y\}]$ is a diamond, contradicting to the assumption that (G, S, k) is reduced with respect to `Small Set Branching`. From $\{x\} \subseteq N_S(B) \cap C$, our claim follows. Notice that $|C \cap B| \leqslant 2$ since an induced cycle can intersect with a clique in at most two vertices. Therefore, $(C \backslash B) \cup \{b\}$ has at least four vertices. Also $G'[(C \backslash B) \cup \{b\}]$ is an induced cycle as no chord can be added in the construction of G' from G. This contradicts to the assumption that $G' - \tilde{S}'$ is a block graph. This completes the proof of the lemma. ◀

▶ **Lemma 6.6.** *Given an instance (G, S, k) to* Disjoint Block Graph Deletion *with $n = |V(G)|$, the algorithm* **Block**(G, S, k) *correctly returns a solution or outputs* No *in time $O(3^{k+\ell} \cdot n^6)$.*

Proof. The correctness of the algorithm is discussed above. For the analysis of the running time, we use the fact that every branching during the execution of **Block**(G, S, k) decreases either k or the number of connected components in $G[S]$ by at least one. The details of the proof can be found in the full version [17]. ◀

Finally, to solve Block Graph Deletion, we apply the standard iterative compression technique. Together with the algorithm **Block** for Disjoint Block Graph Deletion and its analysis given in Lemma 6.6, we obtain an FPT algorithm stated in Theorem 1.2. The full proof is given in the full version [17].

References

1 Hans L. Bodlaender. On disjoint cycles. In *Proceedings of the 17th International Workshop*, WG'91, pages 230–238, London, UK, UK, 1992. Springer-Verlag.

2 Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set: New measure and new structures. *Algorithmica*, pages 1–24, 2014.

3 Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. In *31st International Symposium on Theoretical Aspects of Computer Science*, volume 25 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 214–225. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2014.

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

5 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inform. and Comput.*, 85(1):12–75, 1990.

6 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

7 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time (extended abstract). In *2011 IEEE 52nd Annual Symp. on Foundations of Computer Science (FOCS'11)*, pages 150–159. IEEE CS, 2011.

8 Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM Journal on Discrete Mathematics*, 27(1):290–309, 2013.

9 Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and JakubOnufry Wojtaszczyk. An improved FPT algorithm and quadratic kernel for pathwidth one vertex deletion. In Venkatesh Raman and Saket Saurabh, editors, *Parameterized and Exact Computation*, volume 6478 of *Lecture Notes in Computer Science*, pages 95–106. Springer Berlin Heidelberg, 2010.

10 Frank Dehne, Michael Fellows, Michael Langston, Frances Rosamond, and Kim Stevens. An $O(2^{O(k)}n^3)$ fpt algorithm for the undirected feedback vertex set problem. *Theory of Computing Systems*, 41(3):479–492, 2007.

11 Rod Downey and Michael Fellows. Fixed-parameter tractability and completeness. III. Some structural aspects of the W hierarchy. In *Complexity theory*, pages 191–225. Cambridge Univ. Press, Cambridge, 1993.

12 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-deletion: Approximation, kernelization and optimal FPT algorithms. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 470–479, 2012.

13 T. Gallai. Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen. *Acta Math. Acad. Sci. Hungar.*, 12:131–173, 1961.

14 Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computer and System Sciences*, 72(8):1386–1396, 2006.

15 John Hopcroft and Robert Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, June 1973.

16 Iyad Kanj, Michael Pelsmajer, and Marcus Schaefer. Parameterized algorithms for feedback vertex set. In *Parameterized and Exact Computation*, volume 3162 of *LNCS*, pages 235–247. Springer, 2004.

17 Eun Jung Kim and O-joung Kwon. A polynomial kernel for Block Graph Vertex Deletion, 2015. arXiv.org:abs:1506.08477.

18 Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *Automata, Languages, and Programming – 40th Int'l Colloquium, ICALP 2013, Proceedings, Part I*, pages 613–624, 2013.

19 Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Information Processing Letters*, 114(10):556–560, 2014.

20 Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

21 Neeldhara Misra, Geevarghese Philip, Venkatesh Raman, and Saket Saurabh. On parameterized independent feedback vertex set. *Theoretical Computer Science*, 461(0):65–75, 2012. 17th International Computing and Combinatorics Conference (COCOON 2011).

22 Sang-il Oum. Rank-width and vertex-minors. *J. Combin. Theory Ser. B*, 95(1):79–100, 2005.

23 Venkatesh Raman, Saket Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for undirected feedback vertex set. In *Algorithms and computation*, volume 2518 of *Lecture Notes in Comput. Sci.*, pages 241–248. Springer, Berlin, 2002.

24 Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

25 Neil Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

26 Stéphan Thomassé. A quadratic kernel for feedback vertex set. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 115–119. ACM/SIAM, 2009.