Quick but Odd Growth of Cacti*

Sudeshna Kolay[1], Daniel Lokshtanov[2], Fahad Panolan[1], and Saket Saurabh[1,2]

1 **Institute of Mathematical Sciences, Chennai, India**
2 **University of Bergen, Norway**

Abstract

Let $\mathcal{F}$ be a family of graphs. Given an input graph G and a positive integer k, testing whether G has a k-sized subset of vertices S, such that $G \setminus S$ belongs to $\mathcal{F}$, is a prototype vertex deletion problem. These type of problems have attracted a lot of attention in recent times in the domain of parameterized complexity. In this paper, we study two such problems; when $\mathcal{F}$ is either a family of cactus graphs or a family of odd-cactus graphs. A graph H is called a *cactus* graph if every pair of cycles in H intersect on at most one vertex. Furthermore, a cactus graph H is called an *odd cactus*, if every cycle of H is of odd length. Let us denote by $\mathcal{C}$ and $\mathcal{C}_{\mathsf{odd}}$, families of cactus and odd cactus, respectively. The vertex deletion problems corresponding to $\mathcal{C}$ and $\mathcal{C}_{\mathsf{odd}}$ are called Diamond Hitting Set and Even Cycle Transversal, respectively. In this paper we design randomized algorithms with running time $12^k n^{\mathcal{O}(1)}$ for both these problems. Our algorithms considerably improve the running time for Diamond Hitting Set and Even Cycle Transversal, compared to what is known about them.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Even Cycle Transversal, Diamond Hitting Set, Randomized Algorithms

Digital Object Identifier 10.4230/LIPIcs.IPEC.2015.258

1 Introduction

In the field of parameterized graph algorithms, vertex (edge) deletion (addition, editing) problems constitute a considerable fraction. In particular, let $\mathcal{F}$ be a family of graphs. Given an input graph G and a positive integer k, testing whether G has a k-sized subset of vertices (edges) S, such that $G - S$ belongs to $\mathcal{F}$, is a prototype vertex (edge) deletion problem. Many well known problems in parameterized complexity can be phrased in this language. For example, if $\mathcal{F}$ is a family of edgeless graphs, or forests or bipartite graphs, then it corresponds to Vertex Cover, Feedback Vertex Set, and Odd Cycle Transversal, respectively. Most of these problems are NP-complete due to a classic result by Lewis and Yannakakis [13], and naturally a candidate for parameterized study (with respect to solution size). Vertex Cover, Feedback Vertex Set and Odd Cycle Transversal are some of the most well studied problem in the domain of parameterized complexity. These problems have led to identification of several new techniques and ideas in the field.

Recent years have seen a plethora of results around vertex and edge deletion problems, in the domain of parameterized complexity [3, 4, 8, 9, 10, 11, 12]. In this paper, we continue this line of research and study two vertex deletion problems. In particular we study the

* The research leading to these results has received funding from the European Research Council (ERC) via grants Rigorous Theory of Preprocessing, reference 267959 and PARAPPROX, reference 306992 and the Bergen Research Foundation via grant "Beating Hardness by Preprocessing".

10th International Symposium on Parameterized and Exact Computation (IPEC 2015).
Editors: Thore Husfeldt and Iyad Kanj; pp. 258–269
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

problem of deleting vertices to get a cactus or an odd cactus graph. A graph H is called a *cactus* graph if every pair of cycles in H intersect on at most one vertex. Furthermore, a cactus graph H is called an *odd cactus* graph, if every cycle of H is of odd length. Let us denote by $\mathcal{C}$ and $\mathcal{C}_{\mathsf{odd}}$, families of cacti and odd cacti, respectively. The vertex deletion problems corresponding to $\mathcal{C}$ and $\mathcal{C}_{\mathsf{odd}}$ are called Diamond Hitting Set and Even Cycle Transversal, respectively. It is important to note here that the name of deleting vertices to get into $\mathcal{C}_{\mathsf{odd}}$ is called Even Cycle Transversal, because it is equivalent to deleting a k-sized subset S such that $G - S$ does not have any *cycle of even length*. More precisely, we study the following problems:

Even Cycle Transversal **Parameter:** k
Input: An undirected graph G and a positive integer k.
Question: Does there exist a set S such that $G - S \in \mathcal{C}_{\mathsf{odd}}$?

Diamond Hitting Set **Parameter:** k
Input: An undirected graph G and a positive integer k.
Question: Does there exist a set S such that $G - S \in \mathcal{C}$?

It needs to be mentioned that, in this paper, we refer to multigraphs (may have parallel edges) as graphs. While Odd Cycle Transversal is one of the most well studied problem in the realm of parameterized complexity, there is only one article about Even Cycle Transversal in the literature. The structure of the graph without even cycles, or without cycles 0 modulo some positive integer p, is simple. Thomassen showed that such graphs have treewidth at most $f(p)$ [16]. Misra et al. [15] used the structural properties of an odd-cactus graph to design an algorithm for Even Cycle Transversal with running time $50^k n^{\mathcal{O}(1)}$. They also give an $\mathcal{O}(k^2)$ kernel for the problem. On the other hand the family of cacti $\mathcal{C}$ can be characterised by a single excluded minor. In particular, let Θ be a graph on two vertices that have three parallel edges, then a graph $H \in \mathcal{C}$ if and only if H does not contain Θ as a minor. Since Θ is a connected planar graph we obtain a $c^k n^{\mathcal{O}(1)}$ time algorithm as a corollary to the main results in [8, 11, 12]. It also has $\mathcal{O}(k^2)$ kernel [7]. However, we are not aware of exact value of c as all these algorithms use a protrusion subroutine [2]. In this paper we give the following algorithm for these problems.

▶ **Theorem 1.** *There is a randomised algorithm for* Diamond Hitting Set *and* Even Cycle Transversal *running in time* $12^k n^{\mathcal{O}(1)}$.

Our Methods. Our algorithms use the same methodology that is used for the $4^k n^{\mathcal{O}(1)}$ time algorithm for Feedback Vertex Set [1], and its generalization to Planar $\mathcal{F}$ Deletion [8]. In both our algorithms, we start by applying some reduction rules to the given instance. After this, we show that the number of edges incident to any solution S of our problems, is a constant fraction to the total number of edges in the graph. This counting lemma is our main technical contribution. We also observe that the analysis for the counting lemma is tight for an infinite family of graphs and thus the analysis of our randomized algorithms can not be improved. It is in the same spirit as finding an infinite family of instances for which an approximation algorithm achieves its approximation ratio.

To apply our reduction rules in a way that this fraction is as small as possible, we study a more general problem than Even Cycle Transversal, which we call Parity Even Cycle Transversal. In this problem we are given a graph G and a weight function $w : E(G) \to \{0, 1\}$ and the objective is to delete a subset S of vertices of size at most k such that in $G - S$ there is no cycle whose weight sum is even. Observe that if w assigns one to every edge then it is same as Even Cycle Transversal. We conclude the introduction by noting that Diamond Hitting Set and Even Cycle Transversal admit approximation algorithms with factor 9 and 10 respectively [6, 15].

2 Preliminaries

We denote a graph as G, while its vertex set and edge set as $V(G)$ and $E(G)$ respectively. It is possible that there are parallel edges between two vertices of a graph. The degree of a vertex $v \in V(G)$, denoted by $d_G(v)$, is the number of edges incident on v. The neighbourhood of v, denoted by $N_G(v)$, is the set of vertices that have at least one edge with v. $N_G^2(v)$ is the set of vertices that have a path of length at most two with v. For a subset of vertices S, the subgraph of G induced by S is denoted by $G[S]$. Similarly, for a subset of edges E', the subgraph of G induced by E' is denoted by $G[E']$. For $S \subseteq V(G)$, $G - S$ denotes the induced subgraph $G[V(G) - S]$. Similarly, for $E' \subseteq E(G)$, $G - E'$ denotes the induced subgraph $G[E(G) - E']$. An edge between two vertices $u, v \in V(G)$ is denoted by (u, v), while a path between u, v is denoted by $[u, v]$. If a sequence of vertices $v_1, \ldots, v_t$ or edges $e_1, \ldots e_t$ form a path, then too we denote this path by $[v_1, \ldots, v_t]$ and $[e_1, \ldots e_t]$ respectively. Given two subsets $V_1, V_2 \subseteq V(G)$, $E(V_1, V_2)$ denotes the set of edges in $E(G)$ that have one end point in V_1 and the other in V_2. The subdivision of an edge $e = (u, v)$ of a graph G results in a graph G', which contains a new vertex w, and where the edge e is replaced by two new edges (u, w) and (w, v). A graph $\hat{G}$ is a subdivision of a graph G if there is a sequence of graphs $\{G_1, G_2, \ldots, G_t\}$, with $G_1 = G$ and $G_t = \hat{G}$, where for each $1 < i \leq t$, G_i is obtained by the subdivision of an edge of G_{i-1}.

▶ **Definition 2.** Given a graph G, a cut vertex of G is a vertex v such that $G - \{v\}$ has more components than G. A block of G is a maximal connected subgraph that does not contain any cut vertices of G. A block-decomposition of G is the collection of all blocks. It corresponds to a tree $\mathcal{T}$, where a block X of G corresponds to a vertex t_X of $\mathcal{T}$, and $(t_X, t_Y) \in E(\mathcal{T})$ if the intersection of the corresponding blocks X, Y is exactly one cut vertex.

A block decomposition of a graph can be built in polynomial time.

▶ **Lemma 3** (†). [1] *Let T be a tree. Let $V_1 = \{v \in V(T) \mid d_T(v) = 1\}$, $V_2 = \{v \in V(T) \mid d_T(v) = 2\}$ and $V_3 = \{v \in V(T) \mid d_T(v) \geq 3\}$. Then $\sum_{v \in V_3} d_T(v) \leq 3|V_1|$.*

▶ **Definition 4.** A cactus graph is a connected graph where any two cycles have at most one vertex in common. Equivalently, every edge of the graph belongs to at most one cycle. Another equivalent definition is that a block of a cactus graph can be either a cycle or an edge. A graph where every component is a cactus graph is called a forest of cacti.

▶ **Definition 5.** Let H be a graph on a pair of vertices $\{u, v\}$ that have 3 parallel edges between them. A graph is called a diamond graph if it is obtained by a number of subdivisions of H.

The following Proposition characterizes the class of forests of cacti.

▶ **Proposition 6.** *A graph is a forest of cacti if and only if it does not have a diamond as a subgraph.*

---

[1] Results marked with † can be found in the full version.

The definition of diamond graphs and the characterisation of forests of cacti have been taken from [6]. Please refer to [5] for further details on notations and definitions in Graph Theory.

3 Counting Lemma

In this section, we consider a graph G which has a set S, the deletion of which results in a cactus graph. Moreover, each vertex of the cactus graph has at least three distinct neighbors in G or shares at least two edges with S. Then, it is possible to bound the number of edges in $E(G-S)$ by the number of edges in $E(S, V(G)\setminus S)$. In fact, we exhibit a family of graphs where this bound is tight, up to a constant difference.

▶ **Lemma 7.** *Let G be a graph and $S \subseteq V(G)$ such that $G-S$ is a cactus graph and for all $v \in V(G)\setminus S$ one of the following two conditions holds:*
1. *v has at least 3 distinct neighbors in G, or*
2. *there are at least two edges in $E(v,S)$*
Then $|E(G-S)| \leq 5|E(S, V(G)\setminus S)|$.

Proof. Let $G' = G-S$. We know that G' is a cactus graph. Let $\mathcal{T}$ be the block decomposition tree of G' rooted at a vertex of degree one. Let $B = E(G')$ and $C = E(S, V(G)\setminus S)$ We need to show that $|B| \leq 5|C|$.

Towards the proof, we first define some notations. Let X is a block of size at most 2 (an edge or a cycle of length 2) in G' such that t_X has only one child, which is a leaf node in $\mathcal{T}$. Then we say X and Y together form a *super block*. If blocks X and Y form a super block Z, where t_Y is a leaf node, then by parent of the super block Z, we mean the parent of t_X in $\mathcal{T}$. All other blocks, which are not part of any super block, are called a *normal blocks*. By *size* of a (super/normal) block Z, denoted by $\mathsf{size}(Z)$, we mean the number of edges in the block Z. To bound the number of edges in G' it is enough to bound the total number of edges in super blocks and normal blocks. Let $\mathcal{B}_\ell$ be the set containing all super blocks and normal blocks which correspond to leaves in $\mathcal{T}$. Let $\mathcal{B}_n$ be the set of normal blocks which are not part of $\mathcal{B}_\ell$. Now we define B_ℓ as the set of edges in the (normal/super) blocks which are part of $\mathcal{B}_\ell$, and B_n as the set of edges in the normal blocks which are part of $\mathcal{B}_n$. To bound the cardinality of B, it is enough to bound the cardinality of B_ℓ and B_n, individually. We partition the edges in C as follows. We say an edge $e \in C$ is incident to a (super/normal) block Z if it is incident to a vertex u in Z, which is not the cut vertex shared with the parent of Z. We use E_Z to denote the set of edges in C, which are incident to the (super/normal) block Z. Let C_ℓ be the set of edges in C which are incident to (super/normal) blocks in $\mathcal{B}_\ell$. Similarly, let C_n be the set of edges in C which are incident to blocks in $\mathcal{B}_n$. Let r_i be the number of blocks of size i in $\mathcal{B}_\ell$. Let $B_\ell^{(i)}$ be the set of edges in blocks of size i in $\mathcal{B}_\ell$. Let $C_\ell^{(i)}$ be the set of edges in C_ℓ which are incident to blocks of size i in $\mathcal{B}_\ell$. Notice that $B_\ell = \biguplus_i B_\ell^{(i)}$ and $C_\ell = \biguplus_i C_\ell^{(i)}$.

▶ **Claim 1.** *$r_i \leq \frac{|C_\ell^{(i)}|}{2}$ for $i \leq 4$ and $r_i \leq \frac{|C_\ell^{(i)}|}{i-3}$ for $i \geq 5$.*

Proof.

Bound on r_1. Let X be a block of size one in $\mathcal{B}_\ell$. That is, the block X is a single edge (x, y) and there is a vertex in $\{x, y\}$ which has degree one in G'. Let x be the degree one vertex. By our assumption at least 2 edges in $C_\ell^{(1)}$ are incident on x. This implies that $|E_X| \geq 2$. Thus we have that $|C_\ell^{(1)}| = \sum_{\{X:\mathsf{size}(X)=1\}} E_X \geq 2r_1$. Hence $r_1 \leq \frac{|C_\ell^{(1)}|}{2}$.

Bound on r_2. Let X be a block of size two in $\mathcal{B}_\ell$. If X is a normal block, then the block X is a cycle y, x, y of length 2. Since X is leaf block, there is a vertex in X which is not a cut vertex in G'. Let x be the vertex in X such that x is not a cut vertex. This implies that $N_{G'}(x) = \{y\}$. Thus, by our assumption, either $|E(x, S)| \geq 2$ or x has two neighbors in S. In either case, $|E(x, S)| \geq 2$. That is, $|E_X| \geq 2$. If X is a super block, then X consists of two blocks Y and Z of size 1 each, such that t_Y has only one child t_Z and t_Z is a leaf node in $\mathcal{T}$. Let $Z = (x, y)$ be such that x has degree one in G'. Thus, by our assumption, we can conclude that $|E(x, S)| \geq 2$. That is, $|E_X| \geq 2$. Thus, we have that $|C_\ell^{(2)}| = \sum_{\{X:\mathsf{size}(X)=2\}} E_X \geq 2r_2$. Hence, $r_2 \leq \frac{|C_\ell^{(2)}|}{2}$.

Bound on r_3. Let X be a (super/normal) block of size three in $\mathcal{B}_\ell$. That is, either the block X is a cycle x, y, z, x of length 3, or it is a super block consisting of two blocks, where one of them is a cycle of length 2 and other is an edge. If X is a cycle x, y, z, x, then t_X is a leaf in $\mathcal{T}$. Let z be the only cut vertex in $\{x, y, z\}$. This implies that the degrees of x and y are exactly 2 in G'. Thus, by our assumption, $|E(x, S)| \geq 1$ and $|E(y, S)| \geq 1$. This implies that $|E_X| \geq 2$.

Suppose X is a super block. Then X consists of a cycle x, y, x and an edge (y, z). In this case, only one vertex, either x or z, will be shared with the parent of X and all other vertex will not have a neighbor in $G' - X$. Suppose x is the shared vertex with the parent of the block X. Then the number of distinct neighbors of y and z are exactly 2 and 1 respectively in G'. This implies that $|E(y, S)| \geq 1$ and $|E(z, S)| \geq 2$. Consequently, $|E_X| \geq 3$. By a similar argument, we can show that if z is the shared vertex of the super block X with its parent, then $|E_X| \geq 3$. Thus, we have that $|C_\ell^{(3)}| = \sum_{\{X:\mathsf{size}(X)=3\}} E_X \geq 2r_3$. Hence, $r_3 \leq \frac{|C_\ell^{(3)}|}{2}$.

Bound on r_4. Let X be a (super/normal) block of size four in $\mathcal{B}_\ell$. That is, either the block X is a cycle of length 4 or it is a super block consisting of two blocks. If X is a cycle of length 4, then t_X is a leaf in $\mathcal{T}$. This implies that the degree of every vertex in X, except the cut vertex shared with the parent block, is exactly 2 in G'. This implies that $|E_X| \geq 3$.

Suppose X is a super block consisting of two blocks Y and Z, where size of Y is at most 2 and t_Z is a leaf node in $\mathcal{T}$. If $\mathsf{size}(Y) = 1$, then Z is a cycle of length 3. This implies that at least two vertices in Z has degree exactly 2 in G'. Thus, by our assumption, $|E_Z| \geq 2$ and this implies that $|E_X| \geq 2$.

If $\mathsf{size}(Y) = 2$, then both Y and Z are cycles of length 2. Let x, y, x be the block Y and y, z, y be the block Z. Thus, the number of distinct neighbors of y and z in G' is 2 and 1 respectively. By our assumption, this implies that $|E(y, S)| \geq 1$ and $|E(z, S)| \geq 2$. Thus, we have that $|E_X| \geq 3$. Hence, we conclude that $|C_\ell^{(4)}| = \sum_{\{X:\mathsf{size}(X)=4\}} E_X \geq 2r_4$. This means, $r_4 \leq \frac{|C_\ell^{(4)}|}{2}$.

Bound of r_i for $i \geq 5$. Let X be a (super/normal) block of size at least five in $\mathcal{B}_\ell$. That is, either the block X is a cycle of length i, or it is a super block consisting of two blocks Y and Z such that Z is a cycle of length at least $i - 2$ and t_Z is a leaf in $\mathcal{T}$. In either case, X contains at least $i - 3$ vertices (excluding the cut vertex shared with the parent block) having exactly 2 distinct neighbors in G'. This implies that $|E_X| \geq i - 3$. Hence, we have that $|C_\ell^{(i)}| = \sum_{\{X:\mathsf{size}(X)=i\}} E_X \geq (i - 3)r_i$. Thus, $r_i \leq \frac{|C_\ell^{(i)}|}{i-3}$. ◀

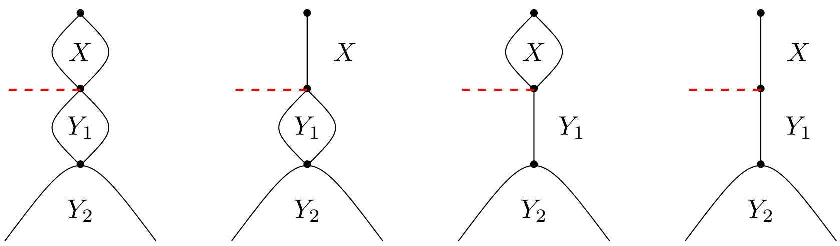

Figure 1 A schematic diagram, when a block X of size at most 2 has only one child which is a super block composed of Y_1 and Y_2. Here the red colored dotted edges belongs to $E(S, V(G) \setminus S)$.

Now we can bound the cardinality of B_ℓ. Let $C_\ell^{(\leq 4)} = \bigcup_{i \leq 4} C_\ell^{(i)}$ and $C_\ell^{(\geq 5)} = \bigcup_{i \geq 5} C_\ell^{(i)}$.

$$|B_\ell| = \sum_i |B_\ell^{(i)}| = \sum_i i \cdot r_i \tag{1}$$

$$\leq 2|C_\ell^{(\leq 4)}| + \sum_{i \geq 5} \frac{i}{i-3} |C_\ell^{(i)}| \qquad \text{(By Claim 1)}$$

$$\leq 2|C_\ell^{(\leq 4)}| + \frac{5}{2}|C_\ell^{(\geq 5)}| \tag{2}$$

What remains is to bound the cardinality of B_n. Let $\mathcal{B}_n^{(\geq 3)}$ be the set of blocks in $\mathcal{B}_n$ such that the corresponding nodes in $\mathcal{T}$ have degree at least 3. That is,

$$\mathcal{B}_n^{(\geq 3)} = \{X \in \mathcal{B}_n \mid d_{\mathcal{T}}(t_X) \geq 3\}.$$

Let $B_n^{(\geq 3)}$ be the set of edges present in the blocks in $\mathcal{B}_n^{(\geq 3)}$. We first bound the cardinality of $B_n^{(\geq 3)}$ and then the cardinality of $B_n \setminus B_n^{(\geq 3)}$. For a set $X \subseteq V(G')$ let numcut_X and $\mathsf{numnoncut}_X$ denote the number of cut vertices and non-cut vertices in X, respectively.

$$|B_n^{(\geq 3)}| = \sum_{X \in \mathcal{B}_n^{(\geq 3)}} |X|$$

$$= \sum_{X \in \mathcal{B}_n^{(\geq 3)}} \mathsf{numcut}_X + \mathsf{numnoncut}_X \tag{3}$$

The quantity $\sum_{X \in \mathcal{B}_n^{(\geq 3)}} \mathsf{numcut}_X$, is at most $\sum_{X \in \mathcal{B}_n^{(\geq 3)}} d_{\mathcal{T}}(t_X)$. This is bounded by three times the number of leaves in $\mathcal{T}$ (by Lemma 3). Thus by Claim 1,

$$\sum_{X \in \mathcal{B}_n^{(\geq 3)}} \mathsf{numcut}_X \leq \frac{3}{2}|C_\ell^{(\leq 4)}| + \frac{3}{2}|C_\ell^{(\geq 5)}| \tag{4}$$

Let $C_n^{\geq 3}$ be the set of edges in C_n which are incident to blocks in $\mathcal{B}_n^{(\geq 3)}$, and $C_n^{\leq 2}$ be the set of edges in C_n which are incident to blocks in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$. For each non-cut vertex x in the block $X \in \mathcal{B}_n^{(\geq 3)}$, there is at least one edge from $C_n^{(\geq 3)}$ which is incident on x. This implies that

$$\sum_{X \in \mathcal{B}_n^{(\geq 3)}} \mathsf{numnoncut}_X \leq |C_n^{(\geq 3)}| \tag{5}$$

Applying Equations 4 and 5 in Equation 3, we get that

$$|B_n^{(\geq 3)}| \leq \frac{3}{2}|C_\ell^{(\leq 4)}| + \frac{3}{2}|C_\ell^{(\geq 5)}| + |C_n^{(\geq 3)}| \tag{6}$$

Now we bound the cardinality of $B_n \setminus B_n^{(\geq 3)}$. First, we bound the number of edges in the blocks in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$ which are not incident to any edge in C_n. Let X be a block in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$,

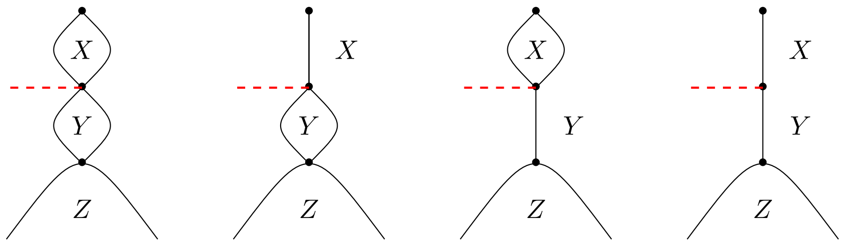

Figure 2 A schematic diagram, when a block X of size at most 2 has only one child Y such that $size(Y) \leq 2$ and $d_{\mathcal{T}}(t_Y) = 2$. Here the red colored dotted edges belongs to $E(S, V(G) \setminus S)$.

such that there is no edge from C_n incident on it. Since t_X has degree 2 in $\mathcal{T}$, the number of cut vertices in X is 2. Now, we claim that $\mathsf{size}(X) \leq 2$. Suppose not. Then there is a vertex x in X such that the degree of x in G' is two. Thus, by our assumption, x is incident to an edge from C_n. This contradicts the fact that there is no edge from C_n is incident on X. Since X is a block in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$, we have that t_X has only one child. Let the child of t_X be t_Y. Now we have the following claim.

▶ **Claim 2.** *Either $d_{\mathcal{T}}(t_Y) \geq 3$ or $Y \in \mathcal{B}_n \setminus \mathcal{B}_n^{(\leq 3)}$ such that there is an edge from $C_n^{(\leq 2)}$ incident on Y.*

Proof. Towards the claim, we first show that $Y \notin \mathcal{B}_\ell$. Suppose not. If Y is a normal block in $\mathcal{B}_\ell$, then X and Y together will form a super block and it contradicts the fact that $X \in \mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$. Suppose Y is a super block in $\mathcal{B}_\ell$. Let Y be the block consisting of blocks Y_1 and Y_2 where t_{Y_2} is a leaf in $\mathcal{T}$ (See Figure 1). Consider the shared vertex x by the blocks X and Y_1. The number of neighbors of x in G' is 2. Thus, by our assumption, x is incident with a vertex in C_n. This contradicts the fact that X be a block in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$ which is not incident to any edge in C_n. Now to prove the claim the only case remaining is $Y \in \mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$, but $d_{\mathcal{T}}(t_Y) = 2$ and there is no edge from $C_n^{(\leq 2)}$ incident on Y (See Figure 2). Then, the size of Y is at most 2. Consider the shared vertex x by the blocks X and Y. The number of neighbors of x in G' is 2. Thus by our assumption x is incident with a vertex in C_n. This contradicts the fact that X be a block in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$ which is not incident to any edge in C_n. This proves the claim. ◀

Using the above claim we can show that the total number of edges in the blocks in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$ which are not incident to any edge in C_n is bounded by

$$2\left(|\mathcal{C}_n^{(\leq 2)}| + \sum_{\{t \in V(\mathcal{T}) : d_{\mathcal{T}}(t) \geq 3\}} 1\right) \leq 2|\mathcal{C}_n^{(\leq 2)}| + 2\sum_i r_i$$
$$\leq 2|\mathcal{C}_n^{(\leq 2)}| + |C_\ell^{(\leq 4)}| + |C_\ell^{(\geq 5)}| \quad \text{(By Claim 1)} \quad (7)$$

Now, we bound the number of edges in the blocks in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$ which are incident to some edges in C_n. Let X be a such a block. If the size of X is at least 3, then there are $i - 2$ vertices in X such that each of these vertices will have only two neighbors in G'. By our assumption, this implies that there are at least $i - 2$ edges from $\mathcal{C}_n^{(\leq 2)}$ which are incident on X. Thus, the total number of edges, in the blocks in $\mathcal{B}_n \setminus \mathcal{B}_n^{(\geq 3)}$, which are not incident to any edge in C_n, is bounded by $3|\mathcal{C}_n^{(\leq 2)}|$. Hence,

$$|B_n \setminus B_n^{(\geq 3)}| = 5|\mathcal{C}_n^{(\leq 2)}| + |C_\ell^{(\leq 4)}| + |C_\ell^{(\geq 5)}| \quad \text{(By Claim 1)} \quad (8)$$

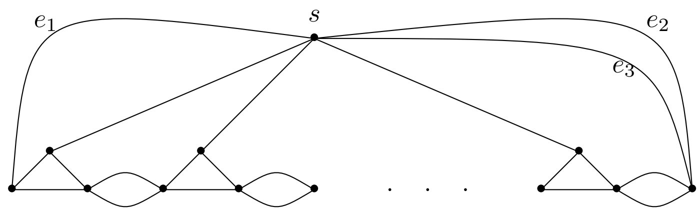

Figure 3 A tight example of Lemma 7. Here $S = \{s\}$.

Hence,

$$\begin{aligned} |B| &= |B_\ell| + |B_n^{(\geq 3)}| + |B_n \setminus B_n^{(\geq 3)}| \\ &= \frac{9}{2}|C_\ell^{(\leq 4)}| + 5|C_\ell^{(\geq 5)}| + 5|\mathcal{C}_n^{(\leq 2)}| + |C_n^{(\geq 3)}| \qquad \text{(By Equations 2,6 and 8)} \\ &\leq 5|C| \end{aligned}$$

This completes the proof of the Lemma. ◀

The bound given in Lemma 7 is in fact tight. Figure 3 represents a family of tight instances. From the figure, let $S = \{s\}$. Let $E_{\mathsf{cross}} = E(S, V(G) \setminus S)$. Let $E' = E_{\mathsf{cross}} - \{e_1, e_2, e_3\}$. Let $E_{\mathsf{cactus}} = E(G - S)$. We see that for every pair of consecutively occuring triangle and double parallel edges in the cactus, there is an edge in E_{cross}. Thus, $|E_{\mathsf{cactus}}| = 5(|E'|)$. This means that $|E_{\mathsf{cactus}}| = 5(|E_{\mathsf{cross}}| - 3)$. Hence, this is a family of tight instances.

4 Algorithm for Even Cycle Transversal

In this section, we give a randomized FPT algorithm for Even Cycle Transversal. This problem is a special case of the following problem.

Parity Even Cycle Transversal	**Parameter:** k
Input: A graph G, a weight function $w : E(G) \to \{0,1\}$ and positive integer k	
Question: Is there a set $S \subseteq V(G)$ of size k such that $G - S$ does not contain any cycle C with $\Sigma_{e \in E(C)} w(e) = 0 \mod 2$?	

We call a cycle C an even-parity (odd-parity) cycle if $\Sigma_{e \in E(C)} w(e) = 0 \mod 2$ ($\Sigma_{e \in E(C} w(e) = 1 \mod 2$). For compactness of notation, we define the function $\mathsf{parity} : 2^{E(G)} \to \{0,1\}$, where for an edge set $E' \in E(G)$, $\mathsf{parity}(E') = \Sigma_{e \in E'} w(e) \mod 2$. In other words, for an even-parity (odd-parity) cycle C, $\mathsf{parity}(E(C)) = 0$ ($\mathsf{parity}(E(C)) = 1$). This should not be confused with cycles of even (odd) length, since we will refer to these cycles simply as even and odd cycles.

In what follows, we give a randomized FPT algorithm for Parity Even Cycle Transversal, that runs in $12^k n^{\mathcal{O}(1)}$ time. First, we preprocess the input graph by applying some reduction rules. A reduction rule reduce an instance (I_1, k) of a problem Π to another instance (I_2, k') of Π. The reduction rule is *safe* when (I_1, k) is a Yes instance if and only if (I_2, k') is a Yes instance. We describe the reduction rules below and prove their safeness. We apply the following rules exhaustively.

▶ **Reduction Rule 1.** *If there is a vertex v in G which is not part of any even-parity cycle, then delete v from G.*

▶ **Lemma 8** (†). *Reduction Rule 1 is safe.*

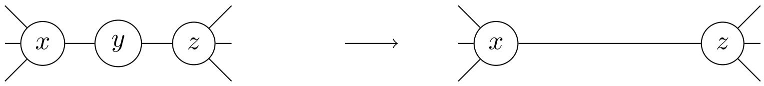

Figure 4 Reduction Rule 2. Here weight of new edge (x, z), $w((x, z)) = (w((x, y)) + w((y, z))$ mod 2.

In the following Lemma, we show that, on a graph where all edges have weight 1, testing whether a vertex is contained in an even cycle can be done in polynomial time.

▶ **Lemma 9.** *Given a graph G, where every edge has weight 1, and a vertex $v \in V(G)$, there is a polynomial time algorithm that checks whether there is an even cycle containing v.*

Proof. The vertex v is contained in an even cycle C if and only if there is a neighbour $u \in N_G(v)$ such that the edge $(u, v) \in E(C)$. For each $u \in N_G(v)$, we check whether there is an even cycle containing the edge (u, v). This is equivalent to checking whether there is an odd path P between v and w in the graph $G' = G - (u, v)$. In [14], the PARITY MULTIWAY CUT (PMWC) problem was posed: If we are given a graph with a set of terminal vertices $T_o \uplus T_e$, does there exist a set S of at most k vertices such that $G - S$ does not have any even path between vertices of T_e and odd paths between vertices of T_o. It was shown that this problem has an FPT algorithm, when parameterised by the size k of the deletion set S. The running time of the algorithm is $2^{2^{\mathcal{O}(k)}} n^{\mathcal{O}(1)}$. We observe that our problem is a special case of the above problem. In our case, $T_o = \{u, v\}, T_e = \emptyset$ and $k = 0$. In other words, we wish to check whether there are any odd paths between u, v in G'. Since $2^{2^{\mathcal{O}(k)}} = \mathcal{O}(1)$, the algorithm for PMWC enables us to check in polynomial time, whether there are no odd paths between u and v in G'. If the algorithm returns YES, then we know that there are no even cycles in G containing the edge (u, v). Otherwise, we conclude that there is an even cycle in G containing v. If, for every edge $e \in E(G)$ adjacent to v, there is no even cycle containing the edge e, then we conclude that there is no even cycle in G that contains v. ◀

This also gives us a polynomial time algorithm to check whether a vertex of a $(0, 1)$ edge-weighted graph is contained in an even-parity cycle.

▶ **Lemma 10** (†). *Given a graph G, where every edge has weight 0 or 1, and a vertex $v \in V(G)$, there is a polynomial time algorithm that checks whether there is an even-parity cycle containing v.*

▶ **Reduction Rule 2.** *Let $[x, y, z]$ be a path in G and degree of y is exactly 2. Then delete y from G and add a new edge $e_1 = (x, z)$. $w(e_1) = w((x, y)) + w((y, z)) \mod 2$. (See Figure 4).*

▶ **Lemma 11.** *Reduction Rule 2 is safe*

Proof. Suppose C is a cycle of parity p in G, which contains the vertex y. Then, since $d_G(y) = 2$, C must contain the path $[x, y, z]$. In the reduced graph G', C is reduced to a cycle C' which contains the edge $e_1 = (x, z)$. By definition of $w(e_1)$, the parity of the reduced cycle is still p. On the other hand, if C' is a cycle of parity p in the reduced graph G', and C' does not contain the new edge e_1, then C' is a cycle of the original graph G. Otherwise, there is a corresponding cycle C in G, which contains the path $[x, y, z]$ instead of the newly added edge e_1. Again, by definition of $w(e_1)$, the parity of C' and C are the same.

Now, suppose (G, k) is a YES instance for PARITY EVEN CYCLE TRANSVERSAL. Let S be a solution set in G. Then S hits all even-parity cycles of G. We have argued that any cycle in G that contains y also contains x and z. Thus, if y was contained in S, then

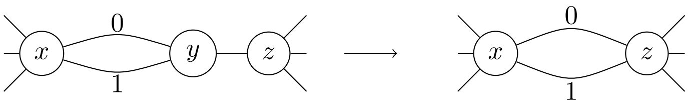

Figure 5 Reduction Rule 3.

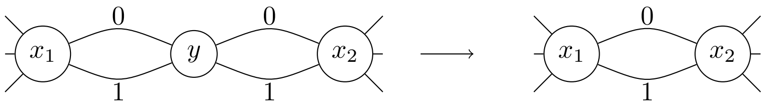

Figure 6 Reduction Rule 4.

$S \cup \{x\} - y$ is also a solution that hits all even-parity cycles of G. Since the parity of cycles is preserved by this reduction, it implies that $S \cup \{x\} - y$ is a solution that hits all even-parity cycles of the reduced graph, and that the reduced instance is also a Yes instance.

On the other hand, suppose the reduced instance is a Yes instance. let S' be a solution set of G'. We will show that S' is also a solution for G. Suppose there is an even-parity cycle C in G, that is not hit by S', then this cycle must have the vertex y. This implies that the cycle must have the path $[x, y, z]$. Let $P = C - \{y\}$. Look at the cycle $C' = P \cup e_1$ in G'. This is also an even-parity cycle which is not hit by S'. This contradicts the fact that S' is a solution set of G'. Thus, (G, k) must be a Yes instance of Parity Even Cycle Transversal. ◀

▶ **Reduction Rule 3.** *Let x, y be two vertices with two parallel edges e_1 and e_2. Let $w(e_1) = 1, w(e_2) = 0$. Further, $e_3 = (y, z)$ is an edge in G, with $z \neq x$, and $d_G(y) = 3$. Then delete y from the graph G and add two new edges $f_1, f_0 = (x, z)$. Define $w(f_1) = 1$ and $w(f_0) = 0$ (See Figure 5).*

▶ **Lemma 12** (†). *Reduction Rule 3 is safe*

▶ **Reduction Rule 4.** *Let $\{x_1, y\}$ be a pair of vertices that have two parallel edges e_1 and e_2, with $w(e_1) = 1, w(e_2) = 0$. Let there be another vertex $x_2 \neq x_1$ such that $\{x_2, y\}$ have two parallel edges e_3 and e_4. It also holds that $w(e_3) = 1, w(e_4) = 0$. Let $d_G(y) = 4$. Then delete y from G and add two new parallel edges f_1, f_0 between x_1 and x_2. We define $w(f_1) = 1$ and $w(f_0) = 0$. (See Figure 6).*

▶ **Lemma 13** (†). *Reduction Rule 4 is safe*

We give the definition of an odd-parity (even-parity) cactus graph and relate it to Parity Even Cycle Transversal.

▶ **Definition 14.** A cactus graph, where the edges have weights from $\{0, 1\}$, is an odd-parity (even-parity) cactus graph when every block of the graph is either an odd-parity (even-parity) cycle or an edge.

▶ **Lemma 15** (†). *Let G be a connected graph and $w : E(G) \to \{0, 1\}$ be a weight function on the edges. G does not contain any cycle C with $w(C) = 0 \mod 2$ if and only if G is an odd-parity cactus.*

Given a graph G, let S be a set of vertices that hits all even-parity cycles. Then each component of $G - S$ does not contain an even-parity cycle. By Lemma 15, it follows that $G - S$ is a forest of odd-parity cacti.

▶ **Observation 1** (†). *Each connected component of the reduced graph for* Parity Even Cycle Transversal *satisfies the conditions of Lemma 7.*

Now, we are ready to describe the algorithm for Parity Even Cycle Transversal.

▶ **Theorem 16.** Parity Even Cycle Transversal *has a randomized algorithm running in* $12^k n^{\mathcal{O}(1)}$ *time.*

Proof. Let S be a solution set of at most k vertices such that $G - S$ is a forest of odd-parity cacti. By Lemma 7, for each component C of G, $|E(C - S)| \leq \frac{5}{6}|E(C \cup S)|$. This implies that $|E(G - S)| \leq \frac{5}{6}|E(G)|$.

Our algorithm is as follows: We define a set $S = \emptyset$ to start with. We pick an edge $e = (u, v) \in E(G)$ uniformly at random and then, with equal probability, we pick one of the two endpoints. We delete this vertex from the current graph and put it into S. In other words, we pick a vertex with probability proportional to its degree. We do this for k steps, at the end of which we check if the constructed set S is a solution set for Parity Even Cycle Transversal. Recognising a forest of odd-parity cacti is equivalent to building a block-decomposition and checking if a block is a odd-parity cycle or an edge. Thus, the entire procedure can be implemented in polynomial time.

Notice that the final set S is a solution set if in each step i, with respect to the current set of vertices in S, we pick a vertex v such that in $G - S$ there is a $k - i$-sized solution set S_i containing v. We will call such a vertex a good vertex for the step i. In step $i \leq k$, the probability, that a good vertex of step i is picked, is at least $\frac{1}{2} \cdot \frac{1}{6} = \frac{1}{12}$. We succeed in finding a solution set S for Parity Even Cycle Transversal if every step picks a good vertex of that step. Thus, the probability of failure in the k-step procedure is at most $1 - (\frac{1}{12})^k$. We repeat the above procedure 12^k times and if in any round we obtain a solution set S of size at most k, we output that set. The probability of failure of this many-round procedure is at most $(1 - (\frac{1}{12})^k)^{12^k} \sim e^{-1}$. The running time of the many-round procedure is $12^k n^{\mathcal{O}(1)}$. ◀

▶ **Corollary 17.** Even Cycle Transversal *has a randomized algorithm running in* $12^k n^{\mathcal{O}(1)}$ *time.*

5 Algorithm for Diamond Hitting Set

In this section, we give a randomized FPT algorithm for Diamond Hitting Set. It was shown in [6] that there is a set of safe reduction rules that can be applied to reduce the input graph to a graph with certain properties.

▶ **Proposition 18** ([6]). *There are polynomial time reduction rules, on application of which, the input instance of* Diamond Hitting Set *is reduced to an equivalent instance where every vertex either has at least three distinct neighbours or three parallel edges.*

▶ **Observation 2** (†). *Each connected component of the reduced graph for* Diamond Hitting Set *satisfies the conditions of Lemma 7.*

Now, we can design an algorithm for Diamond Hitting Set, that is very similar to the algorithm for Parity Even Cycle Transversal.

▶ **Theorem 19** (†). Diamond Hitting Set *has a randomized algorithm running in* $12^k n^{\mathcal{O}(1)}$ *time.*

References

1 Ann Becker, Reuven Bar-Yehuda, and Dan Geiger. Randomized algorithms for the loop cutset problem. *(JAIR)*, 12:219–234, 2000.

2 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. In *FOCS 2009*, pages 629–638, 2009.

3 Yixin Cao. Unit interval editing is fixed-parameter tractable. In *ICALP 2015*, volume 9134 of *LNCS*, pages 306–317. Springer, 2015.

4 Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Transactions on Algorithms*, 11(3):21:1–21:35, 2015.

5 R. Diestel. *Graph Theory.* Springer, Berlin, second ed., electronic edition, February 2000.

6 Samuel Fiorini, Gwenaël Joret, and Ugo Pietropaoli. Hitting diamonds and growing cacti. In *IPCO 2010*, pages 191–204, 2010.

7 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, Geevarghese Philip, and Saket Saurabh. Hitting forbidden minors: Approximation and Kernelization. In Thomas Schwentick and Christoph Dürr, editors, *STACS 2011*, volume 9 of *(LIPIcs)*, pages 189–200, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

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

9 Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. *SIAM J. Comput.*, 42(6):2197–2216, 2013.

10 Archontia C. Giannopoulou, Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. Uniform kernelization complexity of hitting forbidden minors. In *ICALP 2015*, volume 9134 of *LNCS*, pages 629–641. Springer, 2015.

11 Gwenaël Joret, Christophe Paul, Ignasi Sau, Saket Saurabh, and Stéphan Thomassé. Hitting and harvesting pumpkins. *SIAM J. Discrete Math.*, 28(3):1363–1390, 2014.

12 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 *ICALP 2013*, volume 7965 of *LNCS*, pages 613–624. Springer, 2013.

13 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980.

14 Daniel Lokshtanov and M. S. Ramanujan. Parameterized tractability of multiway cut with parity constraints. In *ICALP 2012*, pages 750–761, 2012.

15 Pranabendu Misra, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Parameterized algorithms for even cycle transversal. In *WG 2012*, pages 172–183, 2012.

16 Carsten Thomassen. On the presence of disjoint subgraphs of a specified type. *Journal of Graph Theory*, 12(1):101–111, 1988.