Parameterized Lower Bound and Improved Kernel for Diamond-free Edge Deletion

R. B. Sandeep and Naveen Sivadasan

Department of Computer Science & Engineering
Indian Institute of Technology Hyderabad, India*
{cs12p0001,nsivadasan}@iith.ac.in

Abstract

A diamond is a graph obtained by removing an edge from a complete graph on four vertices. A graph is diamond-free if it does not contain an induced diamond. The Diamond-free Edge Deletion problem asks to find whether there exist at most k edges in the input graph whose deletion results in a diamond-free graph. The problem was proved to be NP-complete and a polynomial kernel of $O(k^4)$ vertices was found by Fellows et. al. (Discrete Optimization, 2011).

In this paper, we give an improved kernel of $O(k^3)$ vertices for Diamond-free Edge Deletion. We give an alternative proof of the NP-completeness of the problem and observe that it cannot be solved in time $2^{o(k)} \cdot n^{O(1)}$, unless Exponential Time Hypothesis fails.

1998 ACM Subject Classification F.2.2. Nonnumerical Algorithms and Problems

Keywords and phrases edge deletion problems, polynomial kernelization

Digital Object Identifier 10.4230/LIPIcs.IPEC.2015.365

1 Introduction

For a finite set of graphs $\mathcal{H}$, $\mathcal{H}$-free Edge Deletion problem asks whether we can delete at most k edges from an input graph G to obtain a graph G' such that for every $H \in \mathcal{H}$, G' does not have an induced copy of H. If $\mathcal{H} = \{H\}$, the problem is denoted by H-free Edge Deletion. $\mathcal{H}$-free Edge Deletion comes under the broader category of graph modification problems which have found applications in DNA physical mapping [3, 13], circuit design [9] and machine learning [2]. Cai has proved that $\mathcal{H}$-free Edge Deletion is fixed parameter tractable [4]. Polynomial kernelization and incompressibility of these problems were subjected to rigorous studies in the recent past. Kratsch and Wahlström gave the first example on the incompressibility of H-free Edge Deletion problems by proving that the problem is incompressible if H is a certain graph on seven vertices, unless NP $\subseteq$ coNP/poly [15]. Later, it has been proved that there exist no polynomial kernel for H-free Edge Deletion where H is any 3-connected graph other than a complete graph, unless NP $\subseteq$ coNP/poly [5]. In the same paper, under the same assumption, it is proved that, if H is a path or a cycle, then H-free Edge Deletion is incompressible if and only if H has at least four edges. It has been proved that $\mathcal{H}$-free Edge Deletion admits polynomial kernelization on bounded degree graphs if $\mathcal{H}$ is a finite set of connected graphs [1]. Though polynomial kernels have been found for many H-free Edge Deletion problems, Claw-free Edge Deletion withstood the test of time and yielded neither an incompressibiltiy result nor a polynomial kernel. Some progress has been made recently for

* The first author is supported by TCS Research Scholarship.

this problem such as a polynomial kernel for Claw-free Edge Deletion on K_t-free input graphs [1] and a polynomial kernel for {Claw, Diamond}-free Edge Deletion [7].

In this paper, we study the polynomial kernelization and parameterized lower bound of Diamond-free Edge Deletion. It has been proved that Diamond-free Edge Deletion is NP-complete and admits a kernel of $O(k^4)$ vertices [11][1]. We improve this result by giving a kernel of $O(k^3)$ vertices. We use vertex modulator technique, which was used recently to give a polynomial kernel for Trivially Perfect Editing [8] and to obtain a polynomial kernel for {Claw, Diamond}-free Edge Deletion [7]. We introduce a rule named as *Vertex Split* which *splits* a vertex into a set of independent vertices where each vertex in the set corresponds to a component in the neighborhood of the vertex. We believe that this rule may have further applications in similar settings.

We give an alternative proof of the NP-completeness of Diamond-free Edge Deletion. Our reduction is from the Vertex Cover problem on cubic graphs and is a linear parameterized reduction. This enables us to prove that, unless Exponential Time Hypothesis (ETH) fails, there exists no parameterized subexponential time algorithm (an algorithm which runs in time $2^{o(k)} \cdot n^{O(1)}$) for Diamond-free Edge Deletion.

1.1 Preliminaries

The problem we consider in this paper is Diamond-free Edge Deletion: whether there exist at most k edges whose deletion from the input graph results in a graph without any induced diamonds. In the parameterized version, the parameter is k.

Graphs: Every graph considered here is simple, finite and undirected. For a graph G, $V(G)$ and $E(G)$ denote the vertex set and the edge set of G respectively. $N_G(v)$ denotes the (open) neighborhood of a vertex $v \in V(G)$, which is the set of vertices adjacent to v in G. The closed neighborhood of v is denoted by $N_G[v]$ and is defined by $N_G(v) \cup \{v\}$. We remove the subscript when there is no ambiguity about the underlying graph G. A graph $G' = (V', E')$ is called an induced subgraph of a graph G if $V' \subseteq V(G)$, $E' \subseteq E(G)$ and an edge $\{x, y\} \in E(G)$ is in E' if and only if $\{x, y\} \subseteq V'$. For a vertex set $V' \subseteq V(G)$, $G[V']$ denotes the induced subgraph with a vertex set V' of G. A component G' of a graph G is a connected induced subgraph of G such that there is no edge between $V(G')$ and $V(G) \setminus V(G')$. For a set of vertices $V' \subseteq V(G)$, $G - V'$ denotes the graph obtained by removing the vertices in V' and all its incident edges from G. For an edge set $E' \subseteq E(G)$, $G - E'$ denotes the graph obtained by deleting all edges in E' from G. If V' (E') is a singleton set $\{v\}$ ($\{e\}$), we denote the graph $G - V'$ ($G - E'$) by $G - v$ ($G - e$). For an edge set $E' \subseteq E(G)$, $V_{E'}(G)$ denotes the vertices in G incident to the edges in E'. A matching (non-matching) is a set of edges (non-edges) such that every vertex in the graph is incident to at most one edge (non-edge) in the matching (non-matching). Diamond is a graph obtained by deleting an edge from a complete graph on four vertices. A graph G is called diamond-free, if G does not contain any diamond as an induced subgraph. Whenever we mention that $\{a, b, c, d\} \subseteq V(G)$ induces a diamond in G, a and b are degree-3 vertices and c and d are degree-2 vertices. In a diamond, we call the edge between the degree-3 vertices as the *middle* edge.

[1] We came to know about this result only after the acceptance of this paper. During this work, our reference was a kernel of $O(k^5)$ vertices [6].

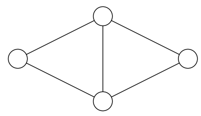


Figure 1 Diamond.

Parameterized complexity: A parameterized problem is *fixed parameter tractable*, if there is an algorithm to solve it in time $f(k) \cdot n^{O(1)}$, where f is any computable function and n is the size of the input, and k is the parameter. A *Polynomial kernelization* is an algorithm which takes as input (G, k), an instance of a parameterized problem, runs in time $(|G|+k)^{O(1)}$ and returns an instance (G', k') of the same problem such that $|G'|, k' \leq p(k)$, where p is any polynomial function. A rule for kernelization is *safe* if (G, k) is a yes-instance if and only if (G', k') is a yes-instance where (G, k) and (G', k') are the input and output of the kernelization. *Linear parameterized reduction* from a parameterized problem A to another B is a polynomial time reduction such that $k' = O(k)$ where k and k' are the parameters of the instances of A and B respectively. A *subexponential time algorithm* for a parameterized problem is an algorithm which runs in time $2^{o(k)} \cdot n^{O(1)}$ where n is the size of the problem instance.

2 Polynomial Kernel

In this section, we give a kernel with $O(k^3)$ vertices for Diamond-free Edge Deletion. Due to space constraints, some of the proofs are deferred to the full version of this paper. To start with, we introduce two properties of graphs and two rules based on those properties.

▶ **Definition 1** (Core Member). A vertex or an edge of a graph G is a *core member* of G if it is a part of some induced diamond or K_4 in G. G has core member property if every vertex and every edge of G is a core member.

▶ **Rule 1** (Irrelevant Edge). *Let (G, k) be an input to the rule. If there is an edge $e \in E(G)$ which is not a core member of G, then delete e from G.*

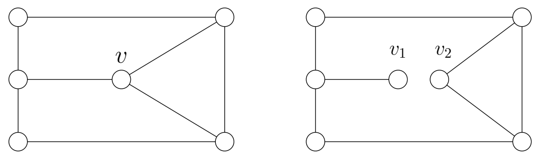
▶ **Lemma 2.** *Irrelevant edge rule is safe and can be applied in polynomial time.*

Proof Idea. An edge which is not a core member is not a part of any minimum solution. Deleting such an edge will not create new diamonds. ◀

▶ **Definition 3** (Connected Neighborhood). For a graph G and a vertex $v \in V(G)$, v has *connected neighborhood* if $G[N(v)]$ is connected. G has connected neighborhood if every vertex in G has connected neighborhood.

▶ **Rule 2** (Vertex-Split). *Let $v \in V(G)$ and v does not have connected neighborhood in G. Let there be $t > 1$ components in $G[N(v)]$ with vertex sets $V_1, V_2, \ldots, V_t$. Introduce t new vertices $v_1, v_2, \ldots, v_t$ and make v_i adjacent to all vertices in V_i for $1 \leq i \leq t$. Delete v.*

An example of the application of vertex-split rule is depicted in Figure 2. We denote the set of vertices created by splitting v by V_v. Let G' be the graph obtained by splitting a vertex v in G. For convenience, we identify an edge (v, u) in G with an edge (v_j, u) in G' where u is in the j^{th} component of $G[N(v)]$, so that for every set of edges S in G, there is a corresponding set of edges in G' and vice versa. We identify a set of vertices $V' \subseteq V(G) \setminus \{v\}$ with the corresponding vertices in G'. Similarly, we identify $V' \subseteq V(G') \setminus V_v$ with the



(a) A graph G where a vertex v is eligible for a split.

(b) Vertex-split rule is applied at v.

Figure 2 An application of vertex-split rule.

corresponding vertex set in G. Before proving the safety of the rule, we prove two simple observations.

▶ **Observation 4.** *Let vertex-split rule be applied on G to obtain G'. Let $v \in V(G)$ be the vertex being split. Then:*

(i) *For every pair of vertices $\{v_i, v_j\} \subseteq V_v$, the distance between v_i and v_j is at least four.*

(ii) *Let $u \in V(G) \setminus \{v\}$ and u has connected neighborhood in G. Then u has connected neighborhood in G'. Furthermore, every new vertex v_i introduced in G' has connected neighborhood.*

Proof. (i). Let $\{v_i, v_j\} \subseteq V_v$. Clearly, v_i and v_j are non-adjacent. Consider any two vertices $u_i \in N(v_i)$ and $u_j \in N(v_j)$. If $u_i = u_j$ or u_i and u_j are adjacent in G', there would be only one vertex generated for the component containing u_i and u_j in $G[N(v)]$ by splitting v, which is a contradiction. It follows that the distance between v_i and v_j is at least four.

(ii). If $v \notin N_G(u)$, then the neighborhood of u is same in both G and G' and hence u has connected neighborhood in G'. Let $v \in N(u)$. Since $G[N_G(u)]$ is connected, to prove that $G'[N_{G'}(u)]$ is connected, it is enough to get an isomorphism between $G[N_G(u)]$ and $G'[N_{G'}(u)]$. Let V' be the set of all vertices in $N_G[u]$ to which v is adjacent. We note that $u \in V'$. Let v_i be the vertex generated by splitting v for the component in $G[N(v)]$ containing u. Since, there is only one new vertex introduced for a component of $G[N(v)]$, no other new vertex is adjacent to u in G'. Now, let V'' be the set of all vertices in $N_{G'}[u]$ to which v_i is adjacent to. Proving $V' = V''$ will establish an isomorphism between $G[N_G(u)]$ and $G'[N_{G'}(u)]$. In order to prove that $V' = V''$ it is enough to prove that $G[V']$ is connected. This is true since $u \in V'$ and u is adjacent to all other vertices in V'. Since a new vertex is made adjacent to a component in the neighborhood of v, every new vertex v_j in G' has connected neighborhood. ◀

▶ **Lemma 5.** *Vertex-split rule is safe and can be applied in polynomial time.*

Proof Idea. Splitting a vertex neither creates nor introduces a new diamond. It does not affect the propagation of the diamonds due to deleting edges. ◀

The next rule deletes an edge e, if e is the middle edge of $k + 1$ otherwise edge-disjoint diamonds. This rule is found in [6].

▶ **Rule 3** (Sunflower). *Let (G, k) be an input to the rule. If there is an edge $e = \{x, y\} \in E(G)$ such that $G[N(x) \cap N(y)]$ has a non-matching of size at least $k + 1$, then delete e from G and decrease k by 1.*

▶ **Lemma 6.** *Sunflower rule is safe and can be applied in polynomial time.*

The next rule is a trivial one.

▶ **Rule 4** (Irrelevant component). *Let (G, k) be an input to the rule. If a component of G is diamond-free, then delete the component from G.*

▶ **Lemma 7.** *Irrelevant component rule is safe and can be applied in polynomial time.*

Now, we are ready with the Phase 1 of the kernelization.

Phase 1

Let (G, k) be an input to Phase 1.

- Exhaustively apply rules irrelevant edge, vertex split, sunflower and irrelevant component on (G, k) to obtain (G', k).

▶ **Lemma 8.** *Let (G', k') be obtained by applying Phase 1 on (G, k). Then:*

(i) *G' has core member property.*
(ii) *G' has connected neighborhood.*
(iii) *Every component in G' has an induced diamond.*
(iv) *$|E(G')| \leq |E(G)|$ and $|V(G')| \leq 2|E(G)|$.*

Proof. (i), (ii) and (iii) follow from the fact that irrelevant edge, vertex-split and irrelevant component rules are not applicable on (G', k). (iv) follows from the fact that none of the rules increases the number of edges in the graph. ◀

▶ **Lemma 9.** *Applying Phase 1 is safe and Phase 1 runs in polynomial time.*

Proof Idea. Follows from Lemma 8(iv) and the safety and running time of each rule. ◀

We define a vertex modulator for Diamond-free Edge Deletion similar to that defined for Trivially Perfect Editing [8].

▶ **Definition 10** (D-modulator). Let (G, k) be an instance of Diamond-free Edge Deletion. Let $V' \subseteq V(G)$ be such that $G[V \setminus V']$ is diamond-free. Then, V' is called a D-modulator.

▶ **Lemma 11** ([10]). *A graph G is diamond-free if and only if every edge in G is a part of exactly one maximal clique.*

For a diamond-free graph G, since every edge is in exactly one maximal clique, there is a unique way of partitioning the edges into maximal cliques. For convenience, we call the set of subsets of vertices, where each subset is the vertex set of a maximal clique, as a *maximal clique partitioning*. We note that, one vertex may be a part of many sets in the partitioning.

▶ **Lemma 12.** *Let (G, k) be an instance of* Diamond-free Edge Deletion. *Then, in polynomial time, the edge set X of a maximal set of edge-disjoint diamonds, a D-modulator V_X of size at most $4k$ and a maximal clique partitioning $\mathcal{C}$ of $G[V(G) \setminus V_X]$ can be obtained or it can be declared that (G, k) is a no-instance.*

Proof Idea. V_X is a set of vertices incident to X. $\mathcal{C}$ can be computed by a greedy method. ◀

Let (G, k) be an output of Phase 1. Here onward, we assume that X is an edge set of the maximal set of edge-disjoint diamonds, V_X is a D-modulator, which is the set of vertices incident to X and $\mathcal{C}$ is the maximal clique partitioning of $G[V(G) \setminus V_X]$. Observation 13 directly follows from the maximality of X. Observation 14 is found in Lemma 3.1 of [7]. It was proved there, if G is {claw, diamond}-free, but is also applicable if G is diamond-free.

▶ **Observation 13.** *Every induced diamond in G has an edge $\{a,b\}$ such that $\{a,b\} \in X$.*

▶ **Observation 14.** *Let $C, C' \in \mathcal{C}$ and be distinct. Then:*
(i) $|C \cap C'| \leq 1$.
(ii) *If $v \in C \cap C'$, then there is no edge between $C \setminus \{v\}$ and $C' \setminus \{v\}$.*

Proof. (i). Assume that $x, y \in C \cap C'$. Then the edge $\{x,y\}$ is part of two maximal cliques, which is a contradiction by Lemma 11.

(ii). Let $x \in C \setminus \{v\}$ and $y \in C' \setminus \{v\}$. Let x and y be adjacent. Clearly, $\{x,y\}$ is not part of the clique induced by C. Now, $\{x,v\}$ is part of not only the clique induced by C but also a maximal clique containing x, y and v, which is a contradiction. ◀

▶ **Definition 15** (Local Vertex)**.** Let G be a graph and $C \subseteq V(G)$ induces a clique in G. A vertex v in C is called local to C in G, if $N(v) \subseteq C$.

▶ **Lemma 16.** *Let (G,k) be an instance of* Diamond-free Edge Deletion*. Let C be a clique with at least $2k+2$ vertices in G.*
(i) *Every solution S of size at most k of (G,k) does not contain any edge e where both the end points of e are in C.*
(ii) *Let $C' \subseteq C$ be such that every vertex $v \in C'$ is local to C in G. Every induced diamond with vertex set D in G can contain at most one vertex in C'.*
(iii) *Let $C' \subseteq C$ be such that every vertex $v \in C'$ is local to C in G. Then, it is safe to delete $\min\{|C'|-1, |C|-(2k+2)\}$ vertices of C' in G.*

Proof Idea. (i). Deleting an edge from a large clique will introduce unmanageable number of diamonds. (ii) follows from the properties of local vertices. (iii). In a big clique, retaining a single local vertex and deleting all other local vertices is safe. ◀

We partition $\mathcal{C}$ into three - $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_{\geq 3}$, the sets of vertices of maximal cliques with one, two and three or more vertices respectively. The first in the following observation has been proved in Lemma 3.2 in [7] in the context where $G - V_X$ is {diamond,claw}-free. Here we prove it in the context where $G - V_X$ is diamond-free.

▶ **Observation 17.** *Let $C \in \mathcal{C}$. Then:*
(i) *If there is a vertex $v \in V_X$ such that v is adjacent to at least two vertices in C, then v is adjacent to all vertices in C.*
(ii) *A vertex in $V(G) \setminus (V_X \cup C)$ is adjacent to at most one vertex in C.*

Proof. (i). Let v is adjacent to two vertices in x, y in C but not adjacent to $z \in C$. Then $\{x,y,v,z\}$ induces a diamond such that none of the edges of the diamond is in X.

(ii). Assume that a vertex $u \in V(G) \setminus (V_X \cup C)$ is adjacent to all vertices in C. This contradicts with the fact that C induces a maximal clique in $G - V_X$. Let u be adjacent to at least two vertices $\{a,b\}$ in C and non-adjacent to at least one vertex $v \in C$. Then $\{a,b,u,v\}$ induces a diamond where none of the edges of the diamond is in X. ◀

Consider $C \in \mathcal{C}$. We define three sets of vertices in G based on C.

$$\begin{aligned} A_C &= \{v \in V_X : v \text{ is adjacent to all vertices in } C\} \\ B_C &= \{v \in V(G) \setminus (V_X \cup C) : v \text{ is adjacent to exactly one vertex in } C\} \\ D_C &= \{v \in V_X : v \text{ is adjacent to exactly one vertex in } C\} \end{aligned}$$

For a vertex $v \in C$, let B_v denote the set of all vertices in B_C adjacent to v. Similarly let D_v denote the set of all vertices in D_C adjacent to v.

▶ **Observation 18.** *Let $C \in \mathcal{C}$. Then,*
(i) *The set of vertices in $V(G) \setminus C$ adjacent to at least one vertex in C is $A_C \cup B_C \cup D_C$.*
(ii) *If $|C| > 1$, then A_C induces a clique in G.*
(iii) *For two vertices $u, v \in C$, $B_u \cap B_v = \emptyset$ and $D_u \cap D_v = \emptyset$.*

Proof. (i) directly follows from Observation 17.

(ii). Assume not. Let a and b be two non-adjacent vertices in A_C. By Observation 17(i), both a and b are adjacent to all vertices in C. Consider any two vertices $x, y \in C$. $\{x, y, a, b\}$ induces a diamond with no edge in X, which is a contradiction.

(iii) directly follows from the definition of B_C and D_C. ◀

▶ **Lemma 19.** *Let $v \in C \in \mathcal{C}_{\geq 3}$. If B_v is non-empty then D_v is non-empty.*

Proof. Since v has connected neighborhood, $G[N(v)]$ is connected. We observe that $N(v) = A_C \cup B_v \cup D_v \cup (C \setminus \{v\})$. Assume B_v is non-empty. There is no edge between the sets B_v and $C \setminus \{v\}$. Consider a vertex $v_b \in B_v$ adjacent to $A_C \cup D_v$. Assume v_b is not adjacent to D_v. Then v_b must be adjacent to a vertex $v_a \in A_C$. Let v' be any other vertex in C. Then $\{v_a, v, v', v_b\}$ induces a diamond which has no edge intersection with X. Therefore v_b must be adjacent to a vertex in D_v. ◀

▶ **Observation 20.** *Let $C \in \mathcal{C}$. Then there are two adjacent vertices x and y such that $x \in A_C$ and $y \in A_C \cup D_C$.*

Proof.
Case 1: $C = \{v\} \in \mathcal{C}_1$. Since $\{v\} \in \mathcal{C}_1$, v is not adjacent to any vertex in $V(G) \setminus V_X$. Since v is a core member, v is part of an induced diamond or K_4 in G. Hence there exist two adjacent vertices $x, y \in A_C$.

Case 2: $C = \{u, v\} \in \mathcal{C}_2$. Since the edge $\{u, v\}$ is a core member, it is part of some induced diamond or K_4 in G. Let a, b be the other two vertices in an induced diamond or K_4 in which $\{u, v\}$ is a part. If both $a, b \in V(G) \setminus V_X$, then it contradicts with either the maximality of X (if a, b, u and v induce a diamond) or with the fact that $\{u, v\}$ is part of exactly one maximal clique C (if a, b, u and v induce a K_4). Let $a \in V_X$ and $b \in V(G) \setminus V_X$. Then, if a, b, u and v induces a diamond, then it contradicts with the maximality of X. If a, b, u and v induces a K_4, then u, v and b induce a K_3 which contradicts with the fact that $\{u, v\}$ is a part of exactly one maximal clique. Hence $a, b \in V_X$. Since a, b, u, v induce a diamond or a K_4, one of a, b must be adjacent to both u and v and the other vertex must be adjacent to at least one of u and v.

Case 3: $C \in \mathcal{C}_{\geq 3}$. Assume that $|A_C| = 0$. If $B_C \cup D_C = \emptyset$, then by Observation 18(i), the clique C is a component in G. Then, irrelevant component rule is applicable. Hence $B_C \cup D_C$ is non-empty. Consider a vertex $v \in C$ such that $B_v \cup D_v$ is non-empty. Consider $N(v)$. $G[N(v)]$ has at least two components, one from $B_v \cup D_v$ and the other from C, which contradicts with the fact that v has connected neighborhood. Hence, $|A_C| > 0$. Assume $|A_C = \{x\}| = 1$. For a contradiction, assume that $D_C = \emptyset$. Then Lemma 19 implies that B_C is empty. Then x does not have connected neighborhood or $C \cup \{x\}$ induces an irrelevant component, which are contradictions. Hence, D_C is non-empty. If $|A_C| \geq 2$, then we are done by Observation 18(ii). ◀

▶ **Lemma 21.** *Let $C \in \mathcal{C}_{\geq 3}$. Then, the number of vertices in C which are adjacent to at least one vertex in $B_C \cup D_C$ is at most $4k - 1$.*

Proof. By Observation 20, $|A_C| \geq 1$. Since $|V_X| \leq 4k$, $|D_C| \leq 4k-1$. Let C' be the set of vertices in C which are adjacent to $B_C \cup D_C$. For every vertex $v \in C'$, by Lemma 19, if B_v is non-empty, then D_v is non-empty. Since $v \in C'$, if B_v is empty, then also D_v is non-empty. For any two vertices $v, u \in C'$, by Observation 18(iii), $D_u \cap D_v = \emptyset$. Therefore $|C'| \leq |D_C| \leq 4k-1$. ◀

Now, we state the last rule of the kernelization.

▶ **Rule 5** (Clique Reduction). *Let $C \in \mathcal{C}_{\geq 3}$ such that $|C| > 4k$. Let C' be $C \cup A_C$. Let C'' be the set of vertices in C which are local to C'. Then, delete any $|C''|-1$ vertices from C''.*

▶ **Observation 22.** *After the application of clique reduction rule, the number of vertices retained in C is at most $4k$.*

Proof. By Lemma 21, the number of vertices in C which are not local to C' is at most $4k-1$. Hence, the rest of the vertices in C are local to C' in G. If $|C| > 4k$, clique reduction rule retains only one local vertex and delete all other vertices in C local to C'. ◀

▶ **Lemma 23.** *Clique reduction rule is safe and can be applied in polynomial time.*

Now we give the kernelization algorithm.

Kernelization of Diamond-free Edge Deletion

Let (G, k) be the input.

Step 1: Apply Phase 1 on (G, k) to obtain (G_1, k_1).

Step 2: Find X, V_X and $\mathcal{C}$ of G_1. Apply clique reduction rule on (G_1, k_1) to obtain (G', k_1).

Step 3: If neither Step 1 nor Step 2 is applicable on (G', k_1), then return (G', k_1). Otherwise apply the kernelization on (G', k_1).

▶ **Lemma 24.** *The kernelization algorithm is safe and can be applied in polynomial time.*

2.1 Bounding the Kernel Size

In this subsection, we bound the number of vertices in the kernel obtained by the kernelization. Let (G, k) be an instance of Diamond-free Edge Deletion and (G', k') is obtained by the kernelization. Consider an X, V_X and $\mathcal{C}$ of (G', k').

▶ **Lemma 25.** $\sum_{C \in \mathcal{C}_1} |C| = O(k^3)$.

Proof. Since, Phase 1 is not applicable on (G', k'), by Lemma 8(i), every vertex is a core member of G'. Let $\{v\} \in \mathcal{C}_1$. By Observation 20, v must be adjacent to two vertices $x, y \in V_X$ such that x and y are adjacent. Now consider the edge $\{x, y\}$. In the common neighborhood of $\{x, y\}$ there can be at most $2k+1$ vertices v with the property that $\{v\} \in \mathcal{C}_1$ (otherwise sunflower rule applies). Since there are at most $O(k^2)$ edges in $G'[V_X]$, we obtain that the total number of vertices in the singleton sets of $\mathcal{C}$ is $O(k^3)$. ◀

▶ **Lemma 26.**

(i) *Consider any two vertices $x, y \in V_X$. Let $\mathcal{C}' \subseteq \mathcal{C}_2 \cup \mathcal{C}_{\geq 3}$ such that for any $C \in \mathcal{C}'$, $x, y \in A_C$. If $\{x, y\} \in X$ then $|\mathcal{C}'| \leq 2k+1$. If $\{x, y\} \notin X$, then $|\mathcal{C}'| \leq 1$.*

(ii) *Consider any ordered pair of vertices (x, y) in V_X such that x and y are adjacent in G'. Let $\mathcal{C}' \subseteq \mathcal{C}_2 \cup \mathcal{C}_{\geq 3}$ such that for any $C \in \mathcal{C}'$, $x \in A_C$ and $y \in D_C$. If $\{x, y\} \in X$ then $|\mathcal{C}'| \leq 2k+1$. If $\{x, y\} \notin X$, then $|\mathcal{C}'| = 0$.*

Proof. (i). Let $C_a, C_b \in \mathcal{C}'$. By Observation 14(i), $|C_a \cap C_b| \le 1$. If $v \in C_a \cap C_b$, then by Observation 14(ii), there is no edge between $C_a \setminus \{v\}$ and $C_b \setminus \{v\}$. Hence, $\{x, v, a, b\}$ induces a diamond where $a \in C_a \setminus \{v\}$ and $b \in C_b \setminus \{v\}$, which is edge disjoint with X, a contradiction. Hence $C_a \cap C_b = \emptyset$. Now, consider any two vertices $a \in C_a$ and $b \in C_b$. Clearly, $\{x, y, a, b\}$ induces a diamond. Hence, $\{x, y\}$ must be an edge in X, otherwise the diamond is edge disjoint with X, a contradiction. Therefore, if $\{x, y\} \notin X$, $|\mathcal{C}'| \le 1$. Now we consider the case in which $\{x, y\} \in X$. If $|\mathcal{C}'| \ge 2k + 2$, we get at least $k + 1$ diamonds where every two diamonds have the only edge intersection $\{x, y\}$. Then sunflower rule applies, which is a contradiction.

(ii). Let $\mathcal{C}'$ be the set of all $C \in \mathcal{C}_2 \cup \mathcal{C}_{\ge 3}$ such that $x \in A_C$ and $y \in D_C$. Consider any two of them - C_a and C_b. By Observation 14(i), $|C_a \cap C_b| \le 1$. If $v \in C_a \cap C_b$, then by Observation 14(ii), there is no edge between $C_a \setminus \{v\}$ and $C_b \setminus \{v\}$. Let $a \in C_a \setminus \{v\}$ and $b \in C_b \setminus \{v\}$. Then $\{x, v, a, b\}$ induces a diamond which is edge disjoint with X, a contradiction. Hence $C_a \cap C_b = \emptyset$. Let $a, a' \in C_a$ such that a is adjacent to y. Then, if $\{x, y\} \notin X$, $\{x, a, a', y\}$ induces a diamond, which is edge disjoint with X. Therefore, if $\{x, y\} \notin X$, then $|\mathcal{C}'| = 0$. Now we consider the case in which $\{x, y\} \in X$. If $|\mathcal{C}'| \ge 2k + 2$, we get at least $k + 1$ diamonds where every two diamonds have the only edge intersection $\{x, y\}$. Then sunflower rule applies, which is a contradiction. ◀

▶ **Lemma 27.** $\sum_{C \in \mathcal{C}_2 \cup \mathcal{C}_{\ge 3}} |C| = O(k^3)$.

Proof. Consider any two adjacent vertices $x, y \in V_X$. Let $\mathcal{C}'_{xy} \subseteq \mathcal{C}_2 \cup \mathcal{C}_{\ge 3}$ be such that $x, y \in A_C$. Then by Lemma 26(i), if $\{x, y\} \in X$, then $|\mathcal{C}'_{xy}| \le 2k + 1$ and if $\{x, y\} \notin X$, then $|\mathcal{C}'_{xy}| \le 1$. Since there are at most $5k$ edges in X and $O(k^2)$ edges in $G[V_X] \setminus X$, $\bigcup_{\{x,y\} \in E(G[V_X])} \mathcal{C}'_{xy}$ has at most $O(k) \cdot (2k + 1) + O(k^2) = O(k^2)$ maximal cliques. Since every maximal clique has at most $4k$ vertices (by Observation 22), the total number of vertices in those cliques is $O(k^3)$.

Now, let $\mathcal{C}'_{xy} \subseteq \mathcal{C}_2 \cup \mathcal{C}_{\ge 3}$ be such that $x \in A_C$ and $y \in D_C$. Then by Lemma 26(ii), if $\{x, y\} \in X$, then $|\mathcal{C}'_{xy}| \le 2k + 1$ and if $\{x, y\} \notin X$, then $|\mathcal{C}'_{xy}| = 0$. Since there are at most $2 \cdot 5k = 10k$ ordered adjacent pairs of vertices in X, $\bigcup_{\{x,y\} \in E(G[V_X])} \mathcal{C}'_{xy}$ has at most $O(k) \cdot (2k + 1)$ maximal cliques. Since every maximal clique has at most $4k$ vertices (by Observation 22), the total number of vertices in those cliques is $O(k^3)$.

Since, by Observation 20, for every $C \in \mathcal{C}$, there exist two vertices $x \in A_C$ and $y \in A_C \cup D_C$, we have counted every $C \in \mathcal{C}_2 \cup \mathcal{C}_{\ge 3}$. Hence $\sum_{C \in \mathcal{C}_2 \cup \mathcal{C}_{\ge 3}} |C| = O(k^3)$. ◀

▶ **Theorem 28.** *Given an instance (G, k) of* Diamond-free Edge Deletion*, the kernelization gives an instance (G', k) such that $|V(G')| = O(k^3)$ and $k' \le k$ or declares that the instance is a no-instance.*

Proof. None of the rules increases the parameter k. Then, the theorem follows from Lemma 25 and Lemma 27 and the fact that $|V_X| \le 4k$. ◀

3 Parameterized Lower Bound

Exponential Time Hypothesis (ETH) (along with Sparsification Lemma [14]) is an assumption that there is no algorithm which solves 3-SAT in time $2^{o(n+m)}(n + m)^{O(1)}$, where n is the number of variables and m is the number of clauses. We can use linear parameterized reduction from 3-SAT (with parameter $n + m$) to another parameterized problem to show that the latter does not have a subexponential parameterized algorithm, unless ETH fails.

In this section, we give a linear parameterized reduction from Vertex Cover on cubic (i.e., every vertex has degree 3) graphs to Diamond-free Edge Deletion. It has been proved that Vertex Cover is NP-complete on graphs with degree at most three [12] and on cubic planar graphs [16]. The reduction in [16] does not imply that there exists no parameterized subexponential time algorithm for Vertex Cover on cubic graphs. But, by modifying the reduction in [12] by using an insight from the reduction in [16], it can be easily proved that Vertex Cover is NP-complete on cubic graphs and cannot be solved in parameterized subexponential time, unless ETH fails.

▶ **Lemma 29.** *Vertex Cover is NP-complete on cubic graphs and cannot be solved in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

Proof Idea. A reduction from 3-SAT to Vertex Cover is given in [12] such that the input 3-SAT instance with n variables and m clauses is satisfiable if and only if the output graph has a vertex cover of size at most $5m$. The output graph has exactly $3m$ vertices with degree 2 and $6m$ vertices with degree 3. In order to make sure that every vertex has degree 3, we can use a technique used in [16] to convert a degree 2 vertex by a simple structure so that the 3-SAT instance is satisfiable if and only if the resultant graph has a vertex cover of size at most $11m$. ◀

Now we give a linear parameterized reduction from Vertex Cover on cubic graphs to Diamond-free Edge Deletion.

Reduction: Let (G, k) be an instance of Vertex Cover and let G be a cubic graph. We replace each edge uv of G by a path of length 3. For every edge uv, we denote the newly introduced vertices as s_{uv} and s_{vu} where s_{uv} is adjacent to u and s_{vu} is adjacent to v. Let S be the set of all new vertices. For every $u \in V(G)$, S_u denotes the three vertices in S adjacent to u. Make every pair of vertices in S_u adjacent to each other such that the vertices in S_u form a triangle. We introduce a universal vertex w which is adjacent to all the vertices in $V(G) \cup S$. For every edge uv in G, we make sure that the edge $s_{uv}s_{vu}$ is un-deletable by making it part of a large clique such that deleting $s_{uv}s_{vu}$ will create unmanageable number of diamonds. For this purpose we introduce a set $C_{\{u,v\}}$ of $6k$ vertices each of them are adjacent to each other and to both s_{uv} and s_{vu}. This completes the reduction. Let the resultant graph be G'.

For every vertex $u \in V(G)$, by G'_u we denote a subgraph of G' induced by $S_u \cup \{u, w\}$. G' when G is a K_4 is given in Figure 3. We will prove that (G, k) is a yes-instance if and only if $(G', 3k)$ is a yes-instance. Before proving this, we observe some properties of $(G', 3k)$.

▶ **Lemma 30.**
(i) *Let E' be a solution of size at most $3k$ of $(G', 3k)$. Then E' does not contain any edge from the graph induced by $C_{\{u,v\}} \cup \{s_{uv}, s_{vu}\}$.*
(ii) *Every induced diamond in G' contain the vertex w.*

Proof. (i). Let C' be $C_{\{u,v\}} \cup \{s_{uv}, s_{vu}\}$. Let $x, y \in C'$ be such that $e = \{x, y\} \in E'$. Consider any pair of vertices $x', y' \in C' \setminus \{x, y\}$. Clearly $\{x', y', x, y\}$ induces a diamond in $G' - e$. Any other pair of vertices $x'', y'' \in C' \setminus \{x, y\}$ such that $\{x', y'\} \cap \{x'', y''\} = \emptyset$ induces a diamond $\{x'', y'', x, y\}$ which is edge disjoint with that induced by $\{x', y', x, y\}$. There should be at least one edge in E' from every such diamond. Since there are $6k$ vertices in $C' \setminus \{x, y\}$, $|E'| \geq 3k + 1$, which is a contradiction.

(ii). Let H be $G' - w$. We claim that H is diamond-free. For every $u \in V(G)$, we observe that $S_u \cup \{u\}$ forms a maximal clique of H. For every pair of adjacent vertices $\{u, v\}$ in G,

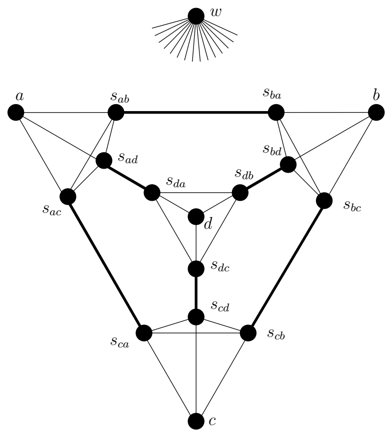


Figure 3 Graph G' when G is a K_4. w is adjacent to all visible vertices. A thick edge $s_{uv}s_{vu}$ denotes a clique of size $6k+2$ with the vertices $C_{\{u,v\}} \cup \{s_{uv}, s_{vu}\}$. s_{uv} and s_{vu} retain its adjacency as shown in the figure, whereas $C_{\{u,v\}}$ vertices are adjacent to only the vertices in $C_{\{u,v\}} \cup \{s_{uv}, s_{vu}\}$.

$C_{\{u,v\}} \cup \{s_{uv}, s_{vu}\}$ forms a maximal clique of H. Now, every edge in H is in one of these maximal cliques. Hence, by Lemma 11, H is diamond-free. ◀

▶ **Theorem 31.** Diamond-free Edge Deletion *is* NP-complete. *Furthermore, the problem cannot be solved in time* $2^{o(k)} \cdot |V(G)|^{O(1)}$*, unless ETH fails.*

Proof. Diamond-free Edge Deletion is trivially in NP. Let (G, k) be an instance of Vertex Cover on cubic graphs and we apply the reduction described to obtain $(G', 3k)$, an instance of Diamond-free Edge Deletion. We need to prove that (G, k) is a yes-instance of Vertex Cover if and only if $(G', 3k)$ is a yes-instance of Diamond-free Edge Deletion.

Let U be a vertex cover of size at most k of G. Let $D = \{s_{uv}w : u \in U, uv \in E(G)\}$, i.e., D is the set of edges between w and S_u for all $u \in U$. We claim that $G' - D$ is diamond-free. To prove this, we give a maximal clique partitioning of $G' - D$. For every vertex $u \in U$, $S_u \cup \{u\}$ is a maximal clique in $G' - D$. For every vertex $v \in V(G) \setminus U$, G'_v is a maximal clique in $G' - D$. For every edge $\{u, v\}$ in G, $C_{\{u,v\}} \cup \{s_{uv}, s_{vu}\}$ is a maximal clique in $G' - D$. Now, we observe that every edge in $G' - D$ is part of some maximal cliques obtained above. Since G is cubic, $|D| \leq 3k$.

Conversely, assume that D is the set of edges in G' such that $G' - D$ is diamond-free and $|D| \leq 3k$. For an edge $\{u, v\}$ in G, $\{s_{uv}, s_{vu}, w, c\}$, where c is any vertex in $C_{\{u,v\}}$ induces a diamond in G'. Since the only deletable edges in this diamond are $s_{uv}w$ and $s_{vu}w$, either of them, say $s_{uv}w$ must be in D. In that case, we observe that at least 2 more edges have to be deleted from G'_u. This implies that, if at all a single edge is deleted from G'_u, then at least 3 edges must be deleted from G'_u. Hence for every edge $uv \in E(G)$ at least 3 edges from G'_u or 3 edges from G'_v must be in D. Now let $U = \{u : D \text{ has an edge from } G'_u\}$. Clearly, U is a vertex cover of size at most k. ◀

4 Concluding Remarks

We obtained an $O(k^3)$ kernel for Diamond-free Edge Deletion which is an improvement over the previously known kernel. We gave an alternative proof for the NP-completeness of Diamond-free Edge Deletion. We observed that the problem cannot be solved in parameterized subexponential time unless ETH fails. We believe that the vertex split rule introduced in this paper will be useful in similar settings. One way of extending our result is to give a polynomial kernel for $\mathcal{H}$-free Edge Deletion where $\mathcal{H}$ is a finite set of graphs containing diamond. We conclude with an open problem: Does Paw-free Edge Deletion admit a polynomial kernel? It is known that a graph is paw-free if and only if every component of it is either triangle-free or complete multipartite [17]. Since this characterization is easier compared to that of claw-free graphs, we believe that finding a polynomial kernel for Paw-free Edge Deletion will be easier compared to that of Claw-free Edge Deletion.

References

1 N. R. Aravind, R. B. Sandeep, and Naveen Sivadasan. On polynomial kernelization of $\mathcal{H}$-free edge deletion. In *Parameterized and Exact Computation*, pages 28–38. Springer International Publishing, 2014.
2 Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.
3 Hans L. Bodlaender and Babette van Antwerpen-de Fluiter. On intervalizing k-colored graphs for DNA physical mapping. *Discrete Applied Mathematics*, 71(1-3):55–77, 1996.
4 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996.
5 Leizhen Cai and Yufei Cai. Incompressibility of H-free edge modification problems. *Algorithmica*, 71(3):731–757, 2015.
6 Yufei Cai. Polynomial kernelisation of H-free edge modification problems. Mphil thesis, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China, 2012.
7 Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, Erik Jan van Leeuwen, and Marcin Wrochna. Polynomial kernelization for removing induced claws and diamonds. In *WG*, 2015.
8 Pål Grønås Drange and Michał Pilipczuk. A polynomial kernel for trivially perfect editing. In *ESA*, 2015.
9 Ehab S El-Mallah and Charles J Colbourn. The complexity of some edge deletion problems. *Circuits and Systems, IEEE Transactions on*, 35(3):354–362, 1988.
10 A Farrugia. Clique-helly graphs and hereditary clique-helly graphs, a mini-survey. *Algoritmic graph theory (CS 762)–2002–Project, Dept. of Comb., University of Waterloo*, 2002.
11 Michael R. Fellows, Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. Graph-based data clustering with overlaps. *Discrete Optimization*, 8(1):2–17, 2011.
12 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete graph problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976.
13 Paul W. Goldberg, Martin Charles Golumbic, Haim Kaplan, and Ron Shamir. Four strikes against physical mapping of DNA. *Journal of Computational Biology*, 2(1):139–152, 1995.
14 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
15 Stefan Kratsch and Magnus Wahlström. Two edge modification problems without polynomial kernels. *Discrete Optimization*, 10(3):193–199, 2013.
16 Bojan Mohar. Face covers and the genus problem for apex graphs. *J. Comb. Theory, Ser. B*, 82(1):102–117, 2001.
17 Stephan Olariu. Paw-fee graphs. *Inf. Process. Lett.*, 28(1):53–54, 1988.