A Polynomial Kernel for Diamond-Free Editing

Yixin Cao
Department of Computing, Hong Kong Polytechnic University, Hong Kong, China
yixin.cao@polyu.edu.hk
https://orcid.org/0000-0002-6927-438X

Ashutosh Rai
Department of Computing, Hong Kong Polytechnic University, Hong Kong, China
ashutosh.rai@polyu.edu.hk

R. B. Sandeep
Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Hungary; and Indian Institute of Technology Dharwad, Dharwad, India
sandeeprb@iitdh.ac.in

Junjie Ye
Department of Computing, Hong Kong Polytechnic University, Hong Kong, China
junjie.ye@polyu.edu.hk
https://orcid.org/0000-0003-3924-008X

Abstract

Given a fixed graph H, the H-free editing problem asks whether we can edit at most k edges to make a graph contain no induced copy of H. We obtain a polynomial kernel for this problem when H is a diamond. The incompressibility dichotomy for H being a 3-connected graph and the classical complexity dichotomy suggest that except for H being a complete/empty graph, H-free editing problems admit polynomial kernels only for a few small graphs H. Therefore, we believe that our result is an essential step toward a complete dichotomy on the compressibility of H-free editing. Additionally, we give a cubic-vertex kernel for the diamond-free edge deletion problem, which is far simpler than the previous kernel of the same size for the problem.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Kernelization, Diamond-free, H-free editing, Graph modification problem

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.10

Funding Supported in part by NSFC under grant 61572414, RGC under grants 152261 and 252026, and ERC under grant 725978.

1 Introduction

A graph modification problem asks whether one can apply at most k modifications to a graph to make it satisfy certain properties. By modifications we usually mean additions and/or deletions, and they can be applied to vertices or edges. Although other modifications are also considered, most results in literature are on vertex deletion and the following three edge modifications: edge deletion, edge completion, and edge editing (deletion/completion).

As usual, we use n to denote the number of vertices of the input graph. For each graph modification problem, one may ask three questions: (1) Is it NP-complete? (2) Can it be solved in time $f(k) \cdot n^{O(1)}$ for some function f, and if yes, what is the (asymptotically) best f? (3) Does it have a polynomial kernel? The first question concerns classic complexity, while the other two are about parameterized complexity [9, 6]. With parameter k, a problem is

26th Annual European Symposium on Algorithms (ESA 2018).
Editors: Yossi Azar, Hannah Bast, and Grzegorz Herman; Article No. 10; pp. 10:1–10:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

fixed-parameter tractable (FPT) if it can be solved in time $f(k) \cdot n^{O(1)}$ for some function f. On the other hand, given an instance (G, k), a *kernelization algorithm* produces in polynomial time an equivalent instance (G', k') – (G, k) is a yes-instance if and only if (G', k') is a yes-instance – such that $k' \leq k$. It is a *polynomial kernel* if the size of G' is bounded from above by a polynomial function of k.

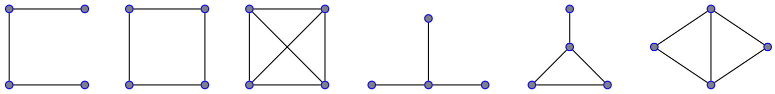
For hereditary properties, a classic result of Lewis and Yannakakis [13] states that all the vertex deletion problems are either NP-hard or trivial. In contrast, the picture for edge modification problems is far murkier. Earlier efforts for edge deletion problems [15, 7], though having produced fruitful concrete results, shed little light on a systematic answer, and it was noted that such a generalization is difficult to obtain.

A basic and ostensibly simple case of graph modification problems is to make the graph H-free, where H is a fixed graph on at least two vertices. (We say that a graph is H-free if it does not contain H as an induced subgraph.) For this special case, all the three questions have been satisfactorily answered for vertex deletion problems, at least in the asymptotic sense. All of them are NP-complete and FPT– indeed, H-free vertex deletion problems admit simple $|V(H)|^k \cdot n^{O(1)}$-time algorithms [2]. On the other hand, the reduction of Lewis and Yannakakis [13] excludes subexponential-time algorithms ($2^{o(k)} \cdot n^{O(1)}$-time algorithms) assuming the exponential time hypothesis (ETH) [11]. Further, as observed by Flum and Grohe [9], the sunflower lemma of Erdős and Rado [8] can be used to produce polynomial kernels for H-free vertex deletion problems.

Even restricted to this very simple case, edge modification problems remain elusive. Significant efforts have been devoted to an ongoing program that tries to answer these questions in a systematic way, and promising progress has been reported in literature. Recently, Aravind et al. [1] gave a complete answer to the first question: The H-free editing problem is NP-complete if and only if H contains at least three vertices. They also excluded subexponential-time algorithms for the NP-complete H-free edge modification problems, assuming ETH. Noting that H-free edge modification problems can always be solved in $2^{O(k)} \cdot n^{O(1)}$ time [2], we are left with the third problem, the existence of polynomial kernels.

Some of the H-free graph classes are important for their own structural reasons, e.g., most notably, cluster graphs and cographs, which are P_3-free graphs and P_4-free graphs respectively; hence the edge modification problems toward them have been well-studied [5, 10]. (Note that edge modification problems to P_2-free graphs, i.e., independent sets, are trivial.) Given the simplicity of H-free edge modification problems, and the naive FPT algorithms for them, it may sound shocking that many of them do *not* admit polynomial kernels. Indeed, the earliest incompressibility results of graph modification problems, by Kratsch and Wahlström [12], are on H-free edge modification problems. Guillemot et al. [10] excluded polynomial kernels for H-free edge deletion problems when H is a path of length at least seven or a cycle of length at least four. An influential result of Cai and Cai [3] furnishes a dichotomy on the compressibility of H-free edge modification problems when H is a path, a cycle, or a 3-connected graph.

We tend to believe that H-free edge modification problems admitting polynomial kernels are the exceptions. Our exploration suggests that graphs on four vertices play the pivotal roles if we want to fully map the territory. Let $\overline{H}$ be the complement graph of H. Then the H-free edge deletion problem is equivalent to the $\overline{H}$-free edge completion problem, while the edge editing problems are the same for H-free and $\overline{H}$-free graphs. We are thus focused on the four-vertex graphs (Figure 1); see Table 1 for a summary of compressibility results of H-free edge modification problems when H is one of them. We conjecture that H-free edge modification problems, when H being claw or paw, admit polynomial kernels.



(a) P_4 **(b)** C_4 **(c)** K_4 **(d)** claw **(e)** paw **(f)** diamond

Figure 1 Graphs on four vertices (their complements are omitted).

Table 1 The compressibility results of H-free edge modification problems for H being four-vertex graphs. Every result holds for the complement H; e.g., the answers are also no when H is $2K_2$.

H	deletion	completion	editing
K_4	$O(k^4)$ [4]	trivial	$O(k^4)$ [4]
P_4	$O(k^3)$ [10]	$O(k^3)$ [10]	$O(k^3)$ [10]
diamond	$O(k^3)$ [14]	trivial	$O(k^8)$ [this paper]
claw	unkown	unkown	unkown
paw	unkown	unkown	unkown
C_4	no [10]	no [10]	no [10]

We show a polynomial kernel for the diamond-free editing problem. Our observations also lead to a cubic-vertex kernel for the diamond-free edge deletion problem, which is far simpler than the previous kernel of the same size [14].

Our key observations are on maximal cliques. A graph G is diamond-free if and only if every two maximal cliques of G share at most one vertex. We say that a maximal clique is of *type* I if it shares an edge with another maximal clique, or *type* II otherwise. It is not hard to see that to make a graph diamond-free, we should never delete edges from a sufficiently large clique. We thus put the maximal cliques of G into three categories, small type I, big type I, and type II. It turns out that a vertex participates in a diamond if and only if it is in a maximal clique of type I, and the small type-I maximal cliques are crucial for the problem.

The first phase of our algorithm comprises two routine reduction rules. If a (non-)edge participates in $k + 1$ or more diamonds that pairwise share only this (non-)edge, then it has to be in a solution. (This is exactly the reason why no edge is deleted from a "large" clique.) If there exists such an edge/non-edge, we delete/add it. We may henceforth assume that these two simple rules have been exhaustively applied. We are able to show that the ends of an edge added by a minimum solution must be from some small type-I maximal cliques. The situation for deleted edges is slightly more complex. The two ends of a deleted edge are either in a small type-I maximal clique, or in a type-II maximal clique. In the second case, the maximal clique has to intersect some small type-I maximal clique.

The second phase of our algorithm uses three nontrivial reduction rules to delete irrelevant vertices. To analyze the size of the kernel, we bound the number of vertices that are (a) in small type-I maximal cliques only, (b) in big type-I maximal cliques but not in any small type-I maximal clique, and (c) only in type-II maximal cliques. First, we show an upper bound on the number of type-I maximal cliques. This immediately bounds the number of vertices in part (a), because each small type-I maximal clique has a bounded size. For part (b), the focus now is to bound the sizes of big maximal cliques of type I. We introduce another reduction rule to delete certain "private vertices" from them. On the other hand, the pattern of vertices shared by big maximal cliques is very limited. We are thus able to bound the number of vertices in part (b), and we are left with part (c). We correlate a maximal clique K of type II with small maximal cliques of type I: we would touch K only because it had become type I after some operation, and this operation has to be an edge addition. Recall that an edge can only be added between two vertices in part (a). For each pair of them, we can build a blocker of $O(k^2)$ vertices from part (c). One more reduction rule is introduced to remove all vertices behind the blockers. Together with the bound of vertices in part (a), this bounds the number of vertices in part (c). They together give our main result.

▶ **Theorem 1.** *The diamond-free editing problem has a kernel of $O(k^8)$ vertices.*

2 Maximal cliques

All graphs discussed in this paper are undirected and simple. A graph G is given by its vertex set $V(G)$ and edge set $E(G)$. The *neighborhood* of a vertex v in a graph G, denoted by $N_G(v)$, consists of all the vertices adjacent to v in G. We extend this to a set $S \subseteq V(G)$ of vertices by defining the neighborhood $N_G(S)$ of S as $(\bigcup_{v\in S} N_G(v)) \setminus S$. For a set $U \subseteq V(G)$ of vertices, we denote by $G[U]$ the subgraph induced by U, whose vertex set is U and whose edge set comprises all edges of G with both ends in U. We use $G - v$, where v is a vertex of G, as a shorthand for $G[V(G) \setminus \{v\}]$. In a diamond, we refer to the edge between the two degree-three vertices as the *cross edge*, and the only non-edge the *missing edge*.

For a set E_+ of edges, we denoted by $G + E_+$ the graph obtained by adding edges in E_+ to G,– its vertex set is still $V(G)$ and its edge set becomes $E(G) \cup E_+$. The graph $G - E_-$ is defined analogously. Throughout the paper we always tacitly assume $E_+ \cap E(G) = \emptyset$ and $E_- \subseteq E(G)$; hence E_+ and E_- are disjoint. A *solution* of an instance (G, k) consists of a set E_+ of added edges and a set E_- of deleted edges such that $G + E_+ - E_-$ is diamond-free and $|E_+ \cup E_-| \leq k$. We use $E_\pm$ as a shorthand for $E_+ \cup E_-$, and there should be no ambiguities: $E_+ = E_\pm \setminus E(G)$ and $E_- = E_\pm \cap E(G)$. We also use $G \triangle E_\pm$ as a shorthand for $G + E_+ - E_-$.

We start from two routine reduction rules for edge editing problems. The correctness of them is straightforward: If we do not add/delete uv, then we have to delete/add at least $k + 1$ edges.

▶ **Rule 1.** *If there exist a non-edge uv and $2k + 2$ distinct vertices $x_1, y_1, \ldots, x_{k+1}, y_{k+1}$ in $N(u) \cap N(v)$ such that $x_i y_i \in E(G)$ for $1 \leq i \leq k + 1$, then add uv and decrease k by one.*

▶ **Rule 2.** *If there exist an edge uv and $2k + 2$ distinct vertices $x_1, y_1, \ldots, x_{k+1}, y_{k+1}$ in $N(u) \cap N(v)$ such that $x_i y_i \notin E(G)$ for $1 \leq i \leq k + 1$, then delete uv and decrease k by one.*

Whether Rule 1 (resp., Rule 2) is applicable to uv can be decided by finding a maximum matching in $G[N(u) \cap N(v)]$ (resp., the complement graph of $G[N(u) \cap N(v)]$). Therefore, Rules 1 and 2 can be applied in polynomial time. We call an instance (G, k) *reduced* if neither Rule 1 nor 2 is applicable to it. In the rest, we will focus on reduced instances. A similar idea as the two rules enables us to exclude some (non-)edges from consideration.

▶ **Proposition 2.** *A (non-)edge uv cannot be in a solution $E_\pm$ of a yes-instance (G, k), if*

(i) *$uv \in E(G)$ and there are $k + 1$ pairwise adjacent vertices in $N(u) \cap N(v)$; or*

(ii) *$uv \notin E(G)$ and there are $k + 1$ pairwise nonadjacent vertices in $N(u) \cap N(v)$.*

▶ **Proposition 3.** *Let (G, k) be a reduced yes-instance. For any (non-)edge uv in a solution of (G, k), the cardinality of $N(u) \cap N(v)$ is at most $3k$.*

Proof. We consider only $uv \in E_-$, and the argument for $uv \in E_+$ is similar and omitted. Let $W = N(u) \cap N(v)$; we find a maximum matching in the complement graph of $G[W]$, and let W' be the ends of the edges in the matching. Since Rule 2 is not applicable (to uv), $|W'| \leq 2k$. There cannot be non-edges between vertices in $W \setminus W'$; then by Proposition 2(i), the size of $W \setminus W'$ is at most k. Therefore, $|W| \leq 3k$. ◀

Our algorithm will be mostly concerned with maximal cliques. According to Proposition 2(i), a maximal clique on $k+3$ or more vertices cannot be touched by a minimum solution "from inside," but it may be touched "from outside". We call a maximal clique *big* if it contains at least $3k+2$ vertices, and *small* otherwise.

▶ **Lemma 4.** *Let (G,k) be a reduced instance.*

(i) *Two big maximal cliques of G share at most one vertex.*

(ii) *If (G,k) is a yes-instance, then a big maximal clique of G remains a maximal clique after applying a solution to (G,k).*

Proof. Let K_1 and K_2 be two big maximal cliques of G. Suppose first that some vertex $u \in K_1 \setminus K_2$ is adjacent to more than $2k+1$ vertices in K_2. Since K_2 is a maximal clique, we can find $v \in K_2 \setminus K_1$ nonadjacent to u, but then Rule 1 would be applicable (to uv). Hence, every vertex in $K_1 \setminus K_2$ has at most $2k+1$ neighbors in K_2, which implies $|K_1 \cap K_2| \le 2k+1$. By assumption, $|K_1| \ge 3k+2$ and $|K_2| \ge 3k+2$. For each vertex in $K_1 \setminus K_2$, we can find $k+1$ non-neighbors in $K_2 \setminus K_1$. Therefore, we can greedily find $k+1$ pairs of distinct vertices $\{x_1, y_1\}$, …, $\{x_{k+1}, y_{k+1}\}$ such that for all $1 \le i \le k+1$, (a) $x_i \in K_1 \setminus K_2$ and $y_i \in K_2 \setminus K_1$; and (b) $x_i y_i \notin E(G)$. Rule 2 would be applicable (to any edge in $G[K_1 \cap K_2]$) if $|K_1 \cap K_2| \ge 2$. Therefore, $|K_1 \cap K_2| \le 1$, and this concludes the proof for assertion (i).

Let $E_\pm$ be a solution to (G,k) and $G^* = G \triangle E_\pm$. By Proposition 2(i), a big maximal clique K in G remains a clique in G^*. Let $v \in V(G) \setminus K$ and let $u \in K \setminus N_G(v)$. Since Rule 1 is not applicable to uv, there are at most $2k+1$ neighbors of v in K. Since $|K| \ge 3k+2$, at least one vertex in K remains nonadjacent to v in G^* because $|E_+| \le k$. Therefore, K is a maximal clique in G^* as well. ◀

It is well known that a graph is diamond-free if and only if every pair of adjacent vertices is contained in exactly one maximal clique. (Proposition 5 implies this fact.) We say that a maximal clique is of *type* I if it shares two or more vertices with some other maximal clique, and *type* II otherwise. We can then rephrase the first sentence of this paragraph as: A graph is diamond-free if and only if it has no type-I maximal clique.

We use $\mathcal{K}_b(G)$, $\mathcal{K}_s(G)$, and $\mathcal{K}_2(G)$ to denote, respectively, the set of big maximal cliques of type I, the set of small maximal cliques of type I, and the set of maximal cliques of type II, of G. A maximal clique in G is in precisely one of them.

▶ **Proposition 5.** *(i) A maximal clique is of type* I *if and only if it contains both ends of the cross edge of a diamond. (ii) A vertex is in a maximal clique of type* I *if and only if it is contained in an induced diamond.*

Proof. The following argument proves assert (i), and it also works for assert (ii).

Let u, v, x, y be four vertices inducing a diamond in G with cross edge uv. We can find two maximal cliques K_1, K_2 containing u, v, x and u, v, y respectively. For any maximal clique K containing u, v, at least one of K_1, K_2 is different from K, hence K is of type I.

We now consider the "only if" direction. Let K_1 be a maximal clique of type I; by definition, there is another maximal clique K_2 such that $|K_1 \cap K_2| \ge 2$. For any vertex $x \in K_1 \setminus K_2$ and any vertex $u \in K_1 \cap K_2$, we can find another vertex $v \in K_1 \cap K_2$ different from u and a vertex $y \in K_2 \setminus K_1$ not adjacent to x (because K_2 is maximal). Clearly, these four vertices induce a diamond with cross edge uv. ◀

The following two statements help us understand edges added by a minimum solution.

▶ **Proposition 6.** *Let G be a diamond-free graph, and let $U \subseteq V(G)$ such that every vertex in $V(G) \setminus U$ is adjacent to at most one vertex of U. If $G[U] \triangle E_\pm$ is diamond-free for a set E_+ of non-edges in $G[U]$ and a set E_- of edges in $G[U]$, then so is $G \triangle E_\pm$.*

Proof. Suppose for contradiction that $G^* = G \triangle E_\pm$ contains a diamond; let D be a set of vertices inducing a diamond in G^*. Since $G[D]$ is not a diamond, at least one (non-)edge of this diamond belongs to $E_\pm$, and is between vertices of U. On the other hand, $G[U] \triangle E_\pm$ remains diamond-free, hence $D \not\subseteq U$. Therefore, $|D \cap U|$ is either two or three, but then a vertex in $D \setminus U$ is adjacent to at least two vertices of $D \cap U$ in G, a contradiction. ◀

▶ **Lemma 7.** *Let $E_\pm$ be a minimum solution to a reduced yes-instance (G, k). Every vertex incident to some edge in E_+ is contained in some small maximal clique of type* I *in G.*

Proof. Let $G^* = G \triangle E_\pm$, where uv is an edge in E_+, and let U be a maximal clique of G^* containing u, v. We argue first that v is in some induced diamond in $G[U]$.

Suppose for contradiction that v participates in no diamond in $G[U]$. Let $X = N_G(v) \cap U$. The subgraph $G[X]$ is a disjoint union of cliques: An induced path of length two would make a diamond with v. Let $\{A_1, \ldots, A_p\}$ be those nontrivial cliques (containing more than one vertex) in $G[X]$; let B be the other vertices of X; and let $C = U \setminus N_G[v]$. Then $\{A_1, \ldots, A_p, B, C\}$ is a partition of the set $U \setminus \{v\}$. Note that p or $|B|$ may be 0, but $|C| > 0$ because $u \in C$. To arrive at a contradiction, we will construct a solution $E'_\pm$ for $G[U]$ whose size is smaller than the number of non-edges in $G[U]$. Assume such an $E'_\pm$ exists and let G' be the graph obtained from G^* by replacing $G^*[U]$ with $G[U] \triangle E'_\pm$. Since U is a type-II maximal clique of G^*, for each $x \in V(G) \setminus U$ we have $|N_{G^*}(x) \cap U| \leq 1$. By Proposition 6, G' is diamond-free. This would however imply a strictly smaller solution than $E_\pm$, contradicting that $E_\pm$ is a minimum solution of (G, k). Now we show how to construct $E'_\pm$.

Case 1, $|B| \geq |C|$. We set $E'_+ = \emptyset$ and E'_- the set of edges in $G[C]$. No edge in E'_- is incident to v or $N(v)$, and hence $N(v) \cap U$ is still a disjoint union of cliques in G'. On the other hand, no vertex $x \in C$ is in any diamond in $G'[U]$ because $N_{G'}(x) \cap U$ is an independent set. Thus, $G'[U]$ is diamond-free. Since B is an independent set of G, and v is nonadjacent to C, we have $|E_+ \cap U^2| \geq \binom{|B|}{2} + |C| \geq \binom{|C|}{2} + |C| > |E'_-| = |E'_+ \cup E'_-|$.

Case 2, $|B| < |C|$. We set E'_+ to be the set of non-edges in $G[B \cup C]$, and E'_- the set of edges between $B \cup C$ and $U \setminus (B \cup C)$. To see that $G'[U]$ is diamond-free, note that its maximal cliques are $B \cup C$ and $\{v\} \cup A_i$ for $1 \leq i \leq p$, whose intersection is either $\{v\}$ or empty. We then calculate the cardinality of $E_+ \cap U^2$, which comprises three parts, those among $B \cup C$, which is exactly E'_+, those between C and v, and those between C and A_i's. Since v does not belong to any diamond in $G[U]$, each vertex in C is adjacent to at most one vertex of $A_i, 1 \leq i \leq p$. In other words, for each $x \in C$ and each $1 \leq i \leq p$, the number of non-edges between x and A_i is at least one. Therefore $|E_+ \cap U^2| \geq |E'_+| + |C| + |C| \times p > |E'_+| + |B| + |C| \times p \geq |E'_+| + |E'_-|$.

Now that v is in some induced diamond in $G[U]$, we can find a maximal clique K of G containing v and another two vertices of this diamond. Since U is not a clique in G, we have $K \neq U$. And $|K \cap U| \geq 3$ implies that K cannot induce a maximal clique of G^*. Hence by Lemma 4(ii), it is small. This concludes the proof of the lemma. ◀

After delimiting the ends of the edges added by a minimum solution, we now turn to the ends of those edges deleted by a minimum solution. The next lemma states that some maximal cliques in G remain maximal cliques after applying the solution.

▶ **Lemma 8.** *Let $E_\pm$ be a minimum solution to an instance (G, k), and let K be a maximal clique of type* II *in G. If E_+ contains neither (i) an edge between $u \in K$ and $v \in N(K)$, nor (ii) two edges between vertices of K and the same vertex in $V(G) \setminus K$, then K remains a maximal clique (of type* II*) in $G \triangle E_\pm$.*

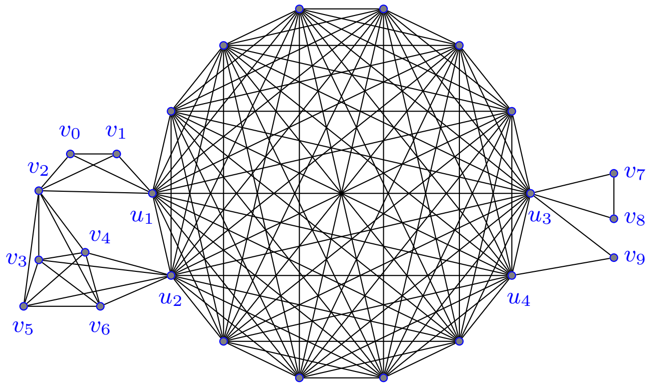


Figure 2 An example with $k = 4$, of which a minimum solution is $\{+u_2v_2, -u_1v_2, -v_0v_1, -u_3v_9\}$. (Note that u_1v_2 and v_0v_1 are not in any diamond of G.) It has six maximal cliques, $K_1 = \{v_0, v_1, v_2, u_1\}$, $K_2 = \{v_2, v_3, v_4, v_5, v_6\}$, $K_3 = \{u_2, v_3, v_4, v_5, v_6\}$, $K_4 = \{u_3, v_7, v_8\}$, $K_5 = \{u_3, u_4, v_9\}$, while K_6 comprises of u_1, u_2, u_3, u_4 and other ten unlabeled vertices. Four of these maximal cliques, K_2, K_3, K_5, and K_6, are of type I, of which only K_6 is big, the other two of type II. All 14 labeled vertices are vulnerable, and the other 8 unlabeled vertices are guarded.

Proof. Let $G^* = G \triangle E_\pm$. Since K is a type-II maximal clique of G, each vertex $v \in V(G) \setminus K$ has at most one neighbor in K. By the assumption that E_+ contains neither (i) nor (ii), this remains true in $G + E_+$ and G^*. On the other hand, E_- cannot contain edges of $G[K]$; otherwise, by Proposition 6, G^* remains diamond-free after replacing $G^*[K]$ by $G[K]$, which implies a strictly smaller solution than $E_\pm$. Therefore, K is a maximal clique in G^*. ◀

The next corollary follows from Lemma 7 and Lemma 8.

▶ **Corollary 9.** *Let $E_\pm$ be a minimum solution to a reduced yes-instance (G, k), and let K be a maximal clique of G containing both ends of an edge in E_-. Then either $K \in \mathcal{K}_s(G)$, or $K \in \mathcal{K}_2(G)$ and K intersects one clique in $\mathcal{K}_s(G)$.*

Lemma 7 and Corollary 9 motivate the following definitions. A vertex v is *vulnerable* in graph G if (1) there exists some $K \in \mathcal{K}_s(G)$ containing v; or (2) there are intersecting maximal cliques $K_1 \in \mathcal{K}_s(G)$ and $K_2 \in \mathcal{K}_2(G)$ such that $v \in K_2$. A vertex is *guarded* if it is not vulnerable. Lemma 7 and Corollary 9 can be summarized as: No (non-)edge in a minimum solution can be incident to a guarded vertex. See Figure 2 for an illustration.

3 The kernel

We partition the vertex set of a reduced graph into five parts, and deal with them separately.

(i) vertices in small maximal cliques of type I (all of them are vulnerable);
(ii) vulnerable vertices in big maximal cliques of type I but not in the previous part;
(iii) other vulnerable vertices (not in any maximal cliques of type I);
(iv) guarded vertices in (big) maximal cliques of type I; and
(v) other guarded vertices (not in any maximal cliques of type I).

Note that for this purpose we do *not* need to enumerate the maximal cliques. The key observation is that we can easily find the cross edges of all diamonds by enumeration, from which we can identify all vertices and edges in maximal cliques of type I. We use the procedure `partition` presented in Algorithm 1, which computes this partition in three steps: It first finds all vertices in a maximal clique of type I, from which it identifies those in a small maximal clique of type I, and finally it uses them to get all vulnerable vertices.

Algorithm 1 The procedure `partition`.

```
INPUT: a reduced instance $(G, k)$.
OUTPUT: vertices in the five parts have (i) mark "small," (ii) marks "vulnerable" and
"type I," (iii) mark "vulnerable," (iv) mark "type I," and (v) no mark, respectively.

1.   for each edge $uv \in E(G)$ where $N(u) \cap N(v)$ does not induce a clique do
1.1.     mark $uv$ "cross edge";
1.2.     mark $u, v$ and all vertices in $N(u) \cap N(v)$ as "type I";
1.3.     mark all edges between these vertices as "type I";
     \\ a vertex is in a maximal clique of type I if and only if it is marked "type I."
2.   for each marked vertex $v$ do
2.1.     if $G[N(v)]$ is not a cluster (a disjoint union of cliques) do mark $v$ "small";
2.2.     else if a clique in $N(v)$ of size $\leq 3k$ contains a cross edge do mark $v$ "small";
3.   for each unmarked edge $uv \in E(G)$ do
3.1.     find the maximal clique $K$ containing $u$ and $v$;
3.2.     if $K$ contains any vertex marked "small" then mark vertices in $K$ "vulnerable";
3.3.     mark every edge in $K$ "checked."
```

It is easy to check that procedure `partition` runs in polynomial time. We now show its correctness.

▶ **Lemma 10.** *Procedure partition is correct.*

Proof. An edge $uv \in E(G)$ is a cross edge if and only if $N(u) \cap N(v)$ does not induce a clique; this justifies step 1.1. Steps 1.2 and 1.3 follow from Proposition 5(i).

Step 2 considers all vertices in maximal cliques of type I. If some component of $G[N(v)]$ is not a clique, we can find a path xyz of length two. There are two different maximal cliques containing v, x, y and v, y, z respectively. Both are of type I, and hence by Lemma 4(i), at least one of them is small. Step 2.2 also follows from Proposition 5(i). If a vertex v is not marked in step 2, then every maximal clique containing v is either big or of type II. Therefore, all vertices in small maximal cliques of type I have been correctly identified in step 2.

Step 3 finds other vulnerable vertices. By definition, such a vertex is in some maximal clique of type II. If a type-II maximal clique consists of an isolated vertex, it is guarded and not marked in step 3. We may hence consider only nontrivial maximal cliques. All edges in a type-II maximal clique remain unmarked. Note that any two vertices of a type-II maximal clique determines this clique: It is the only maximal clique that contains these two vertices. Vertices in the clique are vulnerable if and only if it contains a vertex marked "small." We only need to check the clique K once, so we mark them to avoid unnecessary repetition in step 3.3. After step 3, all type-II maximal cliques have been checked. ◀

3.1 Maximal cliques of type i

We start from the vertices in some small type-I maximal cliques, and denote them by $S(G)$, i.e., $S(G) = \bigcup_{K \in \mathcal{K}_s(G)} K$. Noting that the final graph has no small type-I maximal cliques, we can bound the size of $S(G)$ by relating vertices in it to edges in a minimum solution.

▶ **Lemma 11.** *If (G, k) is a reduced yes-instance, then $|S(G)| \leq 18k^3 + 2k$.*

Proof. Let $E_\pm$ be a minimum solution of (G, k). Let $X = \bigcup_{xy \in E_\pm} \{x, y\}$ and $Y = \bigcup_{xy \in E_\pm} N_G(x) \cap N_G(y)$, i.e., all vertices incident to a (non-)edge in the solution and respectively, all vertices that is a common neighbor of the two ends of a (non-)edge in the solution. Note that $|X| \leq 2k$, and by Proposition 3, $|Y| \leq 3k \cdot |E_{\pm}| \leq 3k^2$. Since $|S(G) \cap (X \cup Y)| \leq |X \cup Y| \leq 3k^2 + 2k$, it suffices to bound $S(G) \setminus (X \cup Y)$. A vertex $v \in S(G) \setminus (X \cup Y)$ cannot be contained in two type-I maximal cliques of G if they share more than one vertex: Otherwise, there is a diamond (as in Proposition 5(ii)) in $N_G[v]$, but then v has to be in $X \cup Y$, a contradiction.

Let us now consider the set of small type-I maximal cliques of G that contain vertices from $S(G) \setminus (X \cup Y)$, which we denote by $\mathcal{K}'$. We argue by contradiction that any pair of cliques in $\mathcal{K}'$ shares at most one vertex. Suppose otherwise, there are two maximal cliques $K_1, K_2 \in \mathcal{K}'$ with $|K_1 \cap K_2| \geq 2$. We have seen that $K_1 \cap K_2$ is disjoint from $S(G) \setminus (X \cup Y)$. Now let $u \in K_1 \setminus K_2$ and $v \in K_2 \setminus K_1$ be two vertices in $S(G) \setminus (X \cup Y)$. Then there is a diamond with u and two vertices in $K_1 \cap K_2$ and one vertex in $K_2 \setminus K_1$. But by the assumption $u \notin X \cup Y$, we cannot add or delete any edge incident to u; on the other hand, $v \notin X \cup Y$ forbids the deletion of other three edges, a contradiction.

Let $v \in S(G) \setminus (X \cup Y)$, and let K be a clique in $\mathcal{K}'$ containing v. By definition, there exists a diamond in which (1) v is a degree-two vertex; (2) the two degree-three vertices are in K; and (3) the other degree-two vertex is not in K. Since v is not in $X \cap Y$, one of the two edges of this diamond that are incident to the other degree-two vertex has to be in E_-. In other words, K contains for some edge $xy \in E_-$, one in $\{x, y\}$ and a common neighbor of x, y. By Proposition 3, for each edge $xy \in E_-$, there are at most $3k$ vertices in $N_G(x) \cap N_G(y)$; for each $z \in N_G(x) \cap N_G(y)$, there can be at most one clique in $\mathcal{K}'$ containing x, z and at most one clique in $\mathcal{K}'$ containing y, z. Therefore, there can be at most $3k \cdot 2 \cdot |E_-| \leq 6k^2$ cliques in $\mathcal{K}'$. By definition, each clique in it is small and has at most $3k + 1$ vertices, of which at least two are not in $S(G) \setminus (X \cup Y)$. Hence $|S(G) \setminus (X \cup Y)| \leq (3k - 1) \cdot 6k^2 = 18k^3 - 6k^2$. Putting the two parts together, we have $|S(G)| \leq 18k^3 + 2k$. ◀

Next, we consider the big type-I maximal cliques, and bound first the number of them.

▶ **Lemma 12.** *If (G, k) is a reduced yes-instance, then $|\mathcal{K}_b(G)| \leq 6k^2$.*

Proof. By Lemma 4, the only way to transform a big maximal clique of type I into one of type II is deleting edges incident to it. For an edge $e = uv \in E_-$, denote by $\mathcal{K}_e$ the set of big type-I maximal cliques containing one in $\{u, v\}$, and one vertex in $N(u) \cap N(v)$. Note that $\mathcal{K}_b(G) = \bigcup_{e \in E_-} \mathcal{K}_e$. By Proposition 3, $\mathcal{K}_e$ has at most $6k$ maximal cliques. Then $|\mathcal{K}_b(G)| \leq 6k \cdot |E_-| = 6k^2$. ◀

To bound the size of big type-I maximal clique, we introduce another reduction rule.

▶ **Rule 3.** *Let $K \in \mathcal{K}_b(G)$ with $|K| \geq 3k + 3$. If K contains a guarded vertex x that does not occur in any other type-I maximal clique of G, delete it.*

▶ **Lemma 13.** *Rule 3 is safe: A reduced instance (G, k) is a yes-instance if and only if $(G - x, k)$ is a yes-instance.*

Proof. It is easy to see that $(G - x, k)$ is a reduced instance, and every solution of (G, k) confined to $G - x$ is a solution of $(G - x, k)$. For the "if" direction, let $E_{\pm}$ be a minimum solution of $(G - x, k)$, and let $G^* = G \triangle E_{\pm}$. Note that $(G - x) \triangle E_{\pm} = G^* - x$, and it is diamond-free. No edge in $E_{\pm}$ is incident to x, and hence $N_G(x) = N_{G^*}(x)$, which we simply denote by $N(x)$. By Proposition 5(ii), it suffices to prove that each maximal clique of G^* containing x is of type II. For this purpose, we show that each component of $G[N(x)]$ is either a single vertex or a type-II maximal clique in $G^* - x$.

Note that $K \setminus \{x\}$ is a big maximal clique in $G - x$: It is a clique of size at least $3k + 2$, and its maximality follows from Lemma 4(i). Hence, by Lemma 4(ii), $K \setminus \{x\}$ is a maximal clique (of type II) in $G^* - x$. Since x is a guarded vertex that does not occur in any other type-I maximal clique, every other maximal clique K' containing x in G is of type II, and it cannot intersect any small type-I maximal clique. Therefore, by Lemma 7, no edge added by E_+ can be incident to any vertex in $N(x)$. From Lemma 8 we can conclude that $K' \setminus \{x\}$ either contains only a vertex or is a maximal clique (of type II) in $G^* - x$.

Since no edge added by E_+ is between two vertices in $N(x)$ and since x is a guarded vertex, each component of $G[N(x)]$ is either $K \setminus \{x\}$ or $K' \setminus \{x\}$, hence is either a single vertex or a type-II maximal clique in $G^* - x$. This concludes the proof. ◀

▶ **Lemma 14.** *Let (G, k) be a reduced yes-instance. If Rule 3 is not applicable, then for each $K \in \mathcal{K}_b(G)$, we have that $|K| = O(k^3)$.*

Proof. Without loss of generality, assume that $|K| \geq 3k + 3$. Since Rule 3 is not applicable, every vertex in K is either a vulnerable vertex, or a guarded vertex in more than one big type-I maximal clique. Let U_1 and U_2 be the set of vulnerable vertices in $K \cap S(G)$ and $K \setminus S(G)$ respectively. By the definition, each vertex in U_2 is adjacent to some vertex in $S(G) \setminus U_1$ by an edge of type-II maximal clique. For each vertex $v \in S(G) \setminus U_1$, the cardinality of $U_2 \cap N(v)$ is at most one; otherwise, there is a type-I maximal clique containing $U_2 \cap N(v)$ and v which by Lemma 4(i) is small, contradicting to $U_2 \subseteq K \setminus S(G)$. Therefore, $|U_2| \leq |S(G) \setminus U_1|$, and by Lemma 11, K contains at most $18k^3 + 2k$ vulnerable vertices. By Lemma 4(i), every pair of big type-I maximal cliques shares at most one vertex. Hence, by Lemma 12, K contains at most $6k^2$ guarded vertices that appear in some other big maximal cliques of type I. Putting them together we get $|K| \leq 18k^3 + 2k + 6k^2$. ◀

The next corollary follows immediately from Lemmas 12 and 14.

▶ **Corollary 15.** *Let (G, k) be a reduced yes-instance. If Rule 3 is not applicable, then the number of vertices that are contained in some cliques in $\mathcal{K}_b(G)$ is $O(k^5)$.*

3.2 Maximal cliques of type ii

We have bounded the number of vertices in all maximal cliques of type I, and it remains to bound the number of vertices that occur *only* in maximal cliques of type II. Let $T(G)$ denote these vertices, i.e., $T(G) = V(G) \setminus \bigcup_{K \in \mathcal{K}_s(G) \cup \mathcal{K}_b(G)} K$. It may not be surprising that we can delete all the guarded vertices in them.

▶ **Rule 4.** *If there is a guarded vertex x not in any type-I maximal clique of G, delete it.*

▶ **Lemma 16.** *Rule 4 is safe: A reduced instance (G, k) is a yes-instance if and only if $(G - x, k)$ is a yes-instance.*

Proof. It is easy to see that $(G - x, k)$ is a reduced instance, and every solution of (G, k) confined to $G - x$ is a solution of $(G - x, k)$. For the other direction, let $E_\pm$ be a minimum solution of $(G - x, k)$, and it is sufficient to show that x is not part of any diamond in $G^* = G \triangle E_\pm$. Note that x is a vertex which is part of only type-II maximal cliques in G and not adjacent to any vertex in small type-I maximal cliques in G. Therefore, by Lemma 7, none of the vertices in $N(x)$ is incident to any edges of E_+. If x is part of a diamond in G^*, then it is formed by a deletion of an edge in $G[N[x]]$ by E_-. But this is not possible by Corollary 9, as none of the edges in $G[N[x]]$ is part of any type-II maximal clique which intersects with a small type-I maximal clique in $G - x$. ◀

If Rule 4 is not applicable, then all vertices in $T(G)$ are vulnerable. As demonstrated in Figure 2, an edge may be deleted from a maximal clique of type II. In that example, neither end of the deleted edge v_0v_1 is in any maximal clique of type I. This can happen only *after* some modification happens in the neighborhood of this vertex – u_2v_2 added in the example. According to Proposition 2, however, this would not happen when $|K| \geq k+3$. In other words, to make sure a large clique in $\mathcal{K}_2(G)$ is immutable to future modifications, it suffices to keep $k+3$ of its vertices. This motivates the following reduction rule, whose statement is however more complex than previous ones. The main trouble here is that we are not allowed to delete all but $k+3$ guarded vertices from a clique in $\mathcal{K}_2(G)$, because it may be required for another clique in $\mathcal{K}_2(G)$. For a pair of vertices u, v, we denote by $N(u, v)$ the set of common neighbors of u and v not in $S(G)$, i.e., $N(u, v) = (N(u) \cap N(v)) \setminus S(G)$.

▶ **Proposition 17.** *Let u, v be two vertices in G. If $uv \notin E(G)$, then $N(u, v)$ form an independent set. Moreover, if $uv \in E_+$ for a solution $E_\pm$ of (G, k), then $|N(u, v)| \leq k$.*

Proof. If $G[N(u, v)]$ has an edge xy, then $\{u, v, x, y\}$ forms a diamond. There are two type-I maximal cliques containing $\{x, y, u\}$ and $\{x, y, v\}$ respectively. By Lemma 4(i), at least one of them is small, contradicting to $x, y \notin S(G)$. The second claim follows from Proposition 2. ◀

Our last rule would keep at most $k+1$ from such sets. To avoid unnecessary clutters, we simply say we mark $k+1$ vertices in $N(u, v)$, even if its size is smaller than $k+1$; in which case, we mark all of them.

▶ **Rule 5.** *For each pair of vertices $u, v \in S(G)$, arbitrarily mark $k+1$ vertices in $N(u, v)$. If $|N(u, v)| \leq k$, then for each vertex $w \in N(u, v)$, arbitrarily mark $k+1$ vertices in $N(u, w)$ and $k+1$ vertices in $N(v, w)$. If there is an unmarked vertex x in $T(G)$, delete it.*

▶ **Lemma 18.** *Rule 5 is safe: A reduced instance (G, k) is a yes-instance if and only if $(G - x, k)$ is a yes-instance.*

Proof. It is easy to see that $(G - x, k)$ is a reduced instance, and every solution of (G, k) confined to $G - x$ is a solution of $(G - x, k)$. For the "if" direction, let $E_\pm$ be a minimum solution of $(G - x, k)$, and let $G^* = G \triangle E_\pm$. We show that each maximal clique of G^* containing x is a maximal clique of G and is of type II in G^*. Since Proposition 5(ii) implies that deleting a vertex not in any type-I maximal clique does not alter type-I maximal cliques, we have $S(G') = S(G)$.

Let K be a maximal clique of G containing x; note that K is a maximal clique of type II in G, as $x \in T(G)$. We argue that $|N_{G^*}(y) \cap K| \leq 1$ for every $y \in V(G) \setminus K$. Since K is a maximal clique of type II in G, we have (1) $|N_G(y) \cap K|$ is either 0 or 1; and (2) for every pair of vertices $u, v \in K$, $N(u, v) \subseteq N_G(u) \cap N_G(v) = K$.

Suppose first that there are at least two edges between y and K in E_+. Let $u, v \in K$ be two vertices such that $yu, yv \in E_+$. Then by Lemma 7, $u, v \in S(G')$, and hence $u, v \in S(G)$. Clearly, $x \neq u$, $x \neq v$ and x is an unmarked vertex in $N(u, v)$. Further, there are $k+1$ marked vertices in $N(u, v)$. It follows that $|K \setminus \{x\}| \geq k+3$, and E_- does not have any edge in $G'[K \setminus \{x\}]$ by Proposition 2(i). Therefore, for each marked vertex $z \in N(u, v)$ that is not adjacent to y, the set $\{u, v, y, z\}$ induces a diamond in $G' + \{yu, yv\}$. The only edge we can edit is yz, but $|N_G(y) \cap K| \leq 1$, and there are at least $k+2$ edges between y and K, which is impossible.

Hence, at most one edge can be added between y and K by E_+. If $|N_G(y) \cap K| = 0$, or $|N_G(y) \cap K| = 1$ but the only edge between y and K is deleted, then it is trivial that y is adjacent to at most one vertex of K in G^*. Suppose that $N_{G^*}(y) \cap K = \{u, v\}$ while only u

is in $N_G(y)$; note that $yu \notin E_-$ and $yv \in E_+$. By Lemma 7, $y, v \in S(G')$, and hence in $S(G)$. According to Proposition 17, there are at most k vertices in $N(v, y)$ in G'. If $u \notin S(G)$, then it has been marked; hence $x \neq u$. Also, $x \neq v$ as $x \in T(G)$. By the rule, no matter whether u is in $S(G)$ or not, we should have marked vertices in $N(u, v)$. Since $x \in N(u, v)$ but is not marked, we have $|N(u, v)| > k + 1$. Let z be any marked vertex in $N(u, v)$; it is not in $N_G(y)$ by assumption. But then $\{u, v, y, z\}$ induces a diamond in $G' + yv$, in which we have to add the missing edge yz, which requires $|E_+| > k$, a contradiction.

We have thus concluded $|N_{G^*}(y) \cap K| \leq 1$ for each vertex y in $V(G) \setminus K$. By Proposition 6, $K \setminus \{x\}$ remains a clique in $G^* - x$, otherwise we can find a strictly smaller solution. Then K is a maximal clique of type II in G^*. On the other hand, according to Proposition 17, no edge is added between two vertices of $N_G(x)$. Therefore, $N(x)$ induces exactly the same subgraph in G and G^*. Hence, any maximal clique of G^* containing x is a maximal clique of G as well, hence of type II in G^*. This concludes the proof of the lemma. ◀

Now Theorem 1 follows by counting numbers of different kinds of vertices.

Proof of Theorem 1. We show first that Rules 3–5 can be applied in polynomial time. For a guarded vertex x, $N(x)$ induces a cluster graph and each maximal clique in the cluster graph together with x forms the maximal cliques of G containing x. Recall that a maximal clique is of type I if and only if it contains both ends of a cross edge. Since the procedure `partition` finds all guarded vertices (no mark) and cross edges, we can find for each guarded vertex all type-I maximal cliques and type-II maximal cliques containing it in polynomial time. Therefore, both Rules 3 and Rule 4 can be applied in polynomial time. Moreover, the procedure `partition` finds all vertices in $S(G)$ (mark "small") and $T(G)$ (no mark "type I"), and hence Rule 5 can be applied in polynomial time.

We claim that if none of Rules 3–5 is applicable to a reduced yes-instance (G, k), then $|V(G)| = O(k^8)$. By Lemma 11, the number of vertices in small type-I maximal cliques is $|S(G)| = O(k^3)$. By Corollary 15, we have $O(k^5)$ vertices in big type-I maximal cliques. For each pair of vertices u, v in $S(G)$, we mark at most $k+1$ common neighbors of them. For each common neighbor w of u, v, we mark at most $2k+2$ vertices: $k+1$ vertices in $N(u, w)$ and $k+1$ vertices in $N(v, w)$. Hence $|T(G)| = O(k^8)$, and $|V(G)| = O(k^3)+O(k^5)+O(k^8) = O(k^8)$. ◀

4 A cubic kernel for diamond-free edge deletion

We use four simple reduction rules to get a cubic kernel for diamond-free edge deletion. The details of this section are omitted due to space limit.

1. If there exist an edge uv and $2k+2$ distinct vertices $x_1, y_1, \ldots, x_{k+1}, y_{k+1}$ in $N(u) \cap N(v)$ such that $x_i y_i \notin E(G)$ for $1 \leq i \leq k+1$, then delete uv and decrease k by one.
2. Mark an edge uv "permanent" if there are $2k+2$ distinct vertices $x_1, y_1, \ldots, x_{k+1}, y_{k+1}$ in $N(u) \cap N(v)$ such that $x_i y_i \in E(G)$ for all $1 \leq i \leq k+1$. If there exists a diamond consisting of only permanent edges, return a trivial no-instance.
3. If there is a vertex x not in any small maximal clique, delete it.
4. Delete all edges and vertices not in any maximal clique of type I.

▶ **Lemma 19.** *Let (G, k) be a yes-instance of the diamond-free edge deletion problem. If none of the reduction rules is applicable, then $|V(G)| = O(k^3)$.*

References

1 N. R. Aravind, R. B. Sandeep, and Naveen Sivadasan. Dichotomy results on the hardness of H-free edge modification problems. *SIAM Journal on Discrete Mathematics*, 31(1):542–561, 2017.

2 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letter*, 58(4):171–176, 1996.

3 Leizhen Cai and Yufei Cai. Incompressibility of H-free edge modification problems. *Algorithmica*, 71(3):731–757, 2015.

4 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.

5 Yixin Cao and Jianer Chen. Cluster editing: Kernelization based on edge cuts. *Algorithmica*, 64(1):152–169, 2012.

6 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Undergraduate texts in computer science. Springer, 2013.

7 Ehab S. El-Mallah and Charles J. Colbourn. The complexity of some edge deletion problems. *IEEE Transactions on Circuits and Systems*, 35(3):354–362, 1988.

8 Paul Erdős and Richard Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, 35(1):85–90, 1960.

9 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006.

10 Sylvain Guillemot, Frédéric Havet, Christophe Paul, and Anthony Perez. On the (non-) existence of polynomial kernels for P_l-free edge modification problems. *Algorithmica*, 65(4):900–926, 2013.

11 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

12 Stefan Kratsch and Magnus Wahlström. Two edge modification problems without polynomial kernels. *Discrete Optimization*, 10(3):193–199, 2013.

13 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

14 R. B. Sandeep and Naveen Sivadasan. Parameterized Lower Bound and Improved Kernel for Diamond-free Edge Deletion. In *10th International Symposium on Parameterized and Exact Computation*, volume 43 of *LIPIcs*, pages 365–376. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

15 Mihalis Yannakakis. Edge-deletion problems. *SIAM Journal on Computing*, 10(2):297–309, 1981.