Theory Comput Syst (2017) 60:615–636
DOI 10.1007/s00224-016-9689-x

Polynomial Kernelization for Removing Induced Claws and Diamonds

Marek Cygan[1] · **Marcin Pilipczuk**[1] ·
Michał Pilipczuk[1] · **Erik Jan van Leeuwen**[2] ·
Marcin Wrochna[1]

Published online: 23 June 2016

Abstract A graph is called {claw, diamond}-free if it contains neither a claw (a $K_{1,3}$) nor a diamond (a K_4 with an edge removed) as an induced subgraph. Equivalently, {claw, diamond}-free graphs are characterized as line graphs of triangle-free graphs, or as linear dominoes (graphs in which every vertex is in at most two maximal cliques and every edge is in exactly one maximal clique). We consider the parameterized complexity of the {claw,diamond}-free Edge Deletion problem, where given a graph G and a parameter k, the question is whether one can remove at most k edges from G to obtain a {claw, diamond}-free graph. Our main result is that this problem admits a polynomial kernel. We complement this result by proving that, even on instances with maximum degree 6, the problem is NP-complete and cannot be solved in time $2^{o(k)} \cdot |V(G)|^{\mathcal{O}(1)}$ unless the Exponential Time Hypothesis fails.

Keywords Edge deletion · Parameterized complexity · Polynomial kernel · {claw, diamond}-free graphs

An extended abstract of this paper will appear in the Proceedings of WG 2015.

The research was supported by Polish National Science Centre grants DEC-2012/05/D/ST6/03214 (Marek Cygan and Marcin Pilipczuk) and DEC-2013/11/D/ST6/03073 (Michał Pilipczuk and Marcin Wrochna). During the work on these results, Marcin Pilipczuk held a postdoc position at the University of Warwick, UK, partially supported by the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick and by the Warwick-QMUL Alliance in Advances in Discrete Mathematics and its Applications; Michał Pilipczuk held a postdoc position at the Warsaw Center of Mathematics and Computer Science, Poland.

✉ Erik Jan van Leeuwen
erikjan@mpi-inf.mpg.de

[1] Institute of Informatics, University of Warsaw, Warsaw, Poland

[2] Max-Planck Institut für Informatik, Saarbrücken, Germany

1 Introduction

Graph modification problems form a wide class of problems, where one is asked to alter a given graph using a limited number of modifications in order to achieve a certain target property, for instance the non-existence of some forbidden induced structures. Depending on the allowed types of modification and the choice of the target property, one can consider a full variety of problems. Well-studied problems that can be expressed in the graph modification paradigm are VERTEX COVER, FEEDBACK VERTEX SET, and CLUSTER EDITING, among others.

It is natural to consider graph modification problems from the parameterized perspective, since they have an innate parameter: the number of allowed modifications, which is expected to be small in applications. As far as the set of allowed modifications is concerned, the most widely studied variants are vertex deletion problems (allowing only removing vertices), edge deletion problems (only removing edges), completion problems (only adding edges), and editing problems (both adding and removing edges). It is very easy to see that as long as the target property can be expressed as the non-existence of induced subgraphs from some finite, fixed list of forbidden subgraphs $\mathcal{F}$ (in other words, belonging to the class of $\mathcal{F}$-free graphs), then all the four variants can be solved in time $c^k \cdot |V(G)|^{\mathcal{O}(1)}$ via a straightforward branching strategy, where the constant c depends on $\mathcal{F}$ only. This observation was first pronounced by Cai [5].

From the perspective of kernelization, again whenever the property is characterized by a finite list of forbidden induced subgraphs, then a standard application of the sunflower lemma gives a polynomial kernel for the vertex deletion variant. The same observation, however, does not carry over to the edge modification problems. The reason is that altering one edge can create new obstacles from $\mathcal{F}$, which need to be dealt with despite not being contained in the original graph G. Indeed, Kratsch and Wahlström [25] have shown a simple graph H on 7 vertices such that the edge deletion problem for the property of being H-free does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly. Later, the same conclusion was proved by Guillemot et al. [20] for H being a long enough path or cycle.

This line of study was continued by Cai and Cai [6] (see also the full version in the master's thesis of Cai [7]), who took up an ambitious project of obtaining a complete classification of graphs H on which edge modification problems for the property of being H-free admit polynomial kernels. The project was very successful: for instance, the situation for 3-connected graphs H is completely understood, and among trees there is only a finite number of remaining unresolved cases. In particular, the study of Cai and Cai revealed that the existence of a polynomial kernel for edge modification problems is actually a rare phenomenon that appears only for very simple graphs H.

One of the most tantalizing questions that is still unresolved is the case $H = K_{1,3}$, i.e., the CLAW-FREE EDGE DELETION problem (as well as the completion and editing variants). The study of this particular case is especially interesting in light of the recent powerful decomposition theorem for claw-free graphs, proved by Chudnovsky and Seymour [8]. For many related problems, having an equivalent structural view on the considered graph class played a crucial role in the design of a polynomial kernel, and hence there is hope for a positive result in this case as well. For this reason, determining the existence of a polynomial kernel for CLAW-FREE EDGE DELETION was posed as an open problem during Workshop on Kernels (WorKer) in 2013, along with the same question for the related LINE GRAPH EDGE DELETION problem [11].

Our Results As an intermediate step towards showing a polynomial kernel for CLAW-FREE EDGE DELETION, we study a related variant, where we forbid *diamonds* as well.[1] By a *diamond* we mean a K_4 with one edge removed, and {claw, diamond}-free graphs are exactly graphs that do not contain claws or diamonds as induced subgraphs. This graph class is equal to the class of line graphs of triangle-free graphs, and to the class of *linear dominoes* (graphs in which every vertex is in at most two maximal cliques and every edge is in exactly one maximal clique) [23, 26].

In this paper, we consider the {CLAW,DIAMOND}-FREE EDGE DELETION problem ({CLAW,DIAMOND}-FREE EDGE DELETION for short) where, given a graph G and an integer k, one is asked to determine whether there exists a subset F of the edges of G with $|F| \leq k$ such that $G - F$ is {claw, diamond}-free.

Our main result is that {CLAW,DIAMOND}-FREE EDGE DELETION admits a polynomial kernel.

Theorem 1 {CLAW,DIAMOND}-FREE EDGE DELETION *admits a polynomial kernel.*

In order to prove Theorem 1, we give a *polynomial-time compression* of {CLAW,DIAMOND}-FREE EDGE DELETION into a problem in NP. By a polynomial-time compression into an unparameterized problem R we mean a polynomial-time algorithm that, given an instance (G, k) of {CLAW,DIAMOND}-FREE EDGE DELETION, outputs an equivalent instance y of R such that $|y| \leq f(k)$, for some computable function f called the *size* of the compression.

Theorem 2 {CLAW,DIAMOND}-FREE EDGE DELETION *admits a polynomial-time compression algorithm into a problem in NP, where the size of the compression is* $\mathcal{O}(k^{24})$.

The problem in NP that Theorem 2 refers to actually is an annotated variant of {CLAW,DIAMOND}-FREE EDGE DELETION. Unfortunately, we are unable to express the annotations in a clean manner using gadgets. Therefore, we compose the polynomial-time compression of Theorem 2 with the NP-hardness reduction that we present for {CLAW,DIAMOND}-FREE EDGE DELETION (see Corollary 1 discussed below) in order to derive Theorem 1.

To prove Theorem 2, we apply the vertex modulator technique. We first greedily pack edge-disjoint claws and diamonds in the input graph. If more than k such obstacles can be packed, then we immediately infer that we are dealing with a

[1] A more detailed discussion of the relation between these two problems is provided in the conclusions section.

no-instance. Otherwise, we obtain a set $X \subseteq V(G)$ with $|X| \leq 4k$ such that every induced claw and diamond in G has at least one edge with both endpoints in X; in particular, $G - X$ is {claw, diamond}-free. This means that we can start to examine the structure of $G - X$ understood as a line graph of a triangle-free graph: it consists of a number of maximal cliques (called henceforth *bags*) that can pairwise share at most a single vertex, and for two intersecting bags B_1, B_2 there is no edge between $B_1 \setminus B_2$ and $B_2 \setminus B_1$. Next, we prove that the neighborhood of every vertex $x \in X$ in $G - X$ is contained only in at most 2 bags, which gives us at most $8k$ bags that are important from the viewpoint of neighborhoods of vertices in X. The crux of the proof lies in observing that an optimum deletion set F consists only of edges that are close to these important bags. Intuitively, all the edges of F lie either in important bags or in bags adjacent to the important ones. A more precise combinatorial analysis leads to a set $S \subseteq V(G)$ of size polynomial in k such that every edge of F has both endpoints in S. After finding such a set S, a polynomial-time compression for the problem can be constructed using a generic argument that works for every edge modification problem with a finite list of forbidden induced subgraphs.

On a high level, our approach uses the vertex modulator technique, which was first used under this name by Drange and Pilipczuk [14] for their recent polynomial kernel for TRIVIALLY PERFECT EDITING. Similar ideas have a long tradition in finding polynomial kernels for packing and deletion problems: see for example Fellows et al. [15] (first Claim of Lemma 2), Brügmann et al. [4], Paul et al. [27] (Lemma 2.2; they use the term "conflict packing"), and many others. However, since we are dealing with a graph class with fundamentally different structural properties from those considered in previous work, the whole combinatorial analysis of the instance with the modulator X (which forms the main part of the paper) is also fundamentally different. We also remark that Cai [7] obtained a kernel for the DIAMOND-FREE EDGE DELETION problem with $\mathcal{O}(k^4)$ vertices. However, the techniques used in that result seem unusable in our setting: their core observation is that a diamond can either be already present in the original graph G or be created by removing an edge of a K_4, and thus one can analyze an auxiliary 'propagation graph' with diamonds and K_4s of the original graph G as nodes. In our setting, we also forbid claws, and the core combinatorial properties of this propagation graph become much too complicated to handle.

Finally, we complement our positive result by proving that {CLAW,DIAMOND}-FREE EDGE DELETION is NP-hard and does not admit a subexponential-time parameterized algorithm unless the Exponential Time Hypothesis of Impagliazzo et al. [22] fails.

Theorem 3 *There exists a polynomial-time reduction that, given an instance ϕ of* 3SAT *with n variables and m clauses, outputs an instance (G, k) of* {CLAW,DIAMOND}-FREE EDGE DELETION *such that (a) (G, k) is a yes-instance if and only if ϕ is satisfiable, (b) $|V(G)|, k = \mathcal{O}(n + m)$, and (c) $\Delta(G) = 6$.*

Corollary 1 *Even on instances with maximum degree 6,* {CLAW,DIAMOND}-FREE EDGE DELETION *is NP-complete and does not admit algorithms with running time $2^{o(k)} \cdot |V(G)|^{\mathcal{O}(1)}$ or $2^{o(|V(G)|)}$ unless the Exponential Time Hypothesis fails.*

Corollary 1 shows that, contrary to recent discoveries for a number of edge modification problems related to subclasses of chordal graphs [2, 3, 13, 18, 19], {CLAW,DIAMOND}-FREE EDGE DELETION does not enjoy the existence of subexponential-time parameterized algorithms. The reduction of Theorem 3 resembles constructions for similar edge modification problems (see e.g. [13, 14, 24]): every variable is replaced by a cyclic variable gadget that has to be completely broken by the solution in one of two possible ways, and variable gadgets are wired together with constant-size clause gadgets that verify the satisfaction of the clauses.

2 Preliminaries

Parameterized Complexity Parameterized complexity is a framework for refining the analysis of a problem's computational complexity by defining an additional "parameter" as part of a problem instance. Formally, a *parameterized problem* is a subset $\mathcal{Q}$ of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. The problem is *fixed-parameter tractable* if there is an algorithm A, a computable function $f : \mathbb{N} \to \mathbb{N}$, and an integer c, such that A solves any instance (x, k) of the problem in time $f(k) \cdot |x|^c$. Note that $f(k)$ does not depend on $|x|$, and that the constant c does not depend on k. The algorithm A is called a *parameterized algorithm*. If $f(k) = 2^{o(k)}$, then we say the algorithm is a subexponential parameterized algorithm.

A *kernelization algorithm* for $\mathcal{Q}$ is an algorithm that takes an instance (x, k) of $\mathcal{Q}$ and in time polynomial in $|x| + k$ outputs an equivalent instance (x', k') (i.e., (x, k) is in $\mathcal{Q}$ if and only if (x', k') is) such that $|x'| \leq g(k)$ and $k' \leq g(k)$ for some computable function g. If the *size* of the kernel g is polynomial, then we say that $\mathcal{Q}$ admits a polynomial kernel. We can relax this definition to the notion of a *compression algorithm*, where the output is required to be an equivalent instance y of some unparameterized problem $\mathcal{Q}'$, i.e., $(x, k) \in \mathcal{Q}$ if and only if $y \in \mathcal{Q}'$. The upper bound $g(k)$ on $|y|$ will then be called the *size* of the compression.

We refer the reader to the books of Downey and Fellows [12], Flum and Grohe [16], and Cygan et al. [10] for a more rigorous introduction.

Graphs We consider finite, undirected, simple graphs G with vertex set $V(G)$ and edge set $E(G)$. Edges $\{u, v\} \in E(G)$ will be written as uv for short. For a subset of vertices $S \subseteq V(G)$, the *subgraph of G induced by S*, denoted $G[S]$, is the graph with vertex set S and edge set $\{uv \in E(G) \mid u, v \in S\}$. We write $G - S$ for $G[V(G) \setminus S]$. For a subset of edges $F \subseteq E(G)$, we write $G - F$ for the subgraph of G obtained by deleting F, that is, $V(G - F) = V(G)$ and $E(G - F) = E(G) \setminus F$. Two disjoint sets $X, Y \subseteq V(G)$ are *fully adjacent* if for every $x \in X$ and $y \in Y$, the vertices x and y are adjacent. If one of these sets is a singleton, say $X = \{v\}$, then we say that v and Y are fully adjacent.

For a vertex $v \in V(G)$, the (open) neighborhood $N_G(v)$ of v is the set $\{u \mid uv \in E(G)\}$. The *closed neighborhood* $N_G[v]$ of v is defined as $N_G(v) \cup \{v\}$. For a subset of vertices $S \subseteq V(G)$, we denote by $E_G(S)$ the set of edges of G with both endpoints in S. In this work N_G and E_G will always pertain to the graph named G, so we drop the subscript.

Cliques, Claws and Diamonds A *clique* of G is a set of vertices that are pairwise adjacent in G; we often identify cliques with the complete subgraphs induced by them. A *maximal clique* is a clique that is not a proper subset of any other clique. A *claw* is a graph on four vertices $\{c, u, v, w\}$ with edge set $\{cu, cv, cw\}$, called *legs* of the claw; we call c the *center* of the claw, and u, v, w the *leaves* of the claw. When specifying the vertices of a claw we always give the center first. A diamond is a graph on four vertices $\{u, v, w, x\}$ with edge set $\{uv, uw, vw, vx, wx\}$.

Forbidden Induced Subgraphs Consider any finite family of graphs $\mathcal{H}$. A graph G is *$\mathcal{H}$-free* if for every $H \in \mathcal{H}$, G does not contain H as an induced subgraph (H is not isomorphic to $G[S]$ for any $S \subseteq V(G)$). An *HDS* (*$\mathcal{H}$-free deletion set*) for G is a subset of edges $F \subseteq E(G)$ such that $G - F$ is $\mathcal{H}$-free. Whenever we talk about a *minimal* HDS, we mean inclusion-wise minimality. $\mathcal{H}$-FREE EDGE DELETION is the parameterized problem asking, for a graph G and a parameter k, whether G has an HDS of size at most k. In ANNOTATED $\mathcal{H}$-FREE EDGE DELETION we are additionally given a set $S \subseteq V(G)$ and the question is whether G has an HDS of size at most k that is contained in $E(S)$.

Let (G, k) be an instance of $\mathcal{H}$-FREE EDGE DELETION. Recall that we can easily find a subset X of the *vertices* of G of size polynomial in k such that (in particular) $G - X$ is $\mathcal{H}$-free. We refer to such a set as a *modulator* of G. The construction here is basically the same as in Lemma 3.3 of [14], and a slightly stronger construction based on the Sunflower Lemma can be found in [17]. In fact, the underlying idea is prevalent in many earlier works on packing and deletion problems: see for example [4, 15, 27].

Lemma 1 *Let $c = \max\{|V(H)| : H \in \mathcal{H}\}$. Then one can in polynomial time either find a subset $X \subseteq V(G)$ of size at most $c \cdot k$ such that every induced $H \in \mathcal{H}$ in G has an edge in $E(X)$, or conclude that (G, k) is a no-instance.*

Proof Let $\{H_1, H_2, \ldots, H_m\}$ be an inclusion-wise maximal set of edge-disjoint induced subgraphs from $\mathcal{H}$ in G (such a set can be greedily found in polynomial time). Since any HDS must contain an edge from each of the subgraphs, it must have size at least m. If $m > k$, then we can conclude that (G, k) is a no-instance. Otherwise, let $X = \bigcup_i V(H_i)$ be the union of their vertex sets; clearly $|X| \leq \sum_{i=1}^{m} |V(H_i)| \leq c \cdot k$. By the maximality of our choice, every induced $H \in \mathcal{H}$ in G has an edge in common with one of $H_1, \ldots, H_m$ and thus an edge in $E(X)$. Hence X satisfies the claim. □

We finish this section by showing that it suffices to find a set S of vertices of size polynomial in k such that every minimal solution (every minimal HDS of size at most k) is contained in $E(S)$. As we show rigorously in Lemma 2 below, given such a set we can compress the $\mathcal{H}$-FREE EDGE DELETION instance in polynomial time to an instance of the annotated version with $\mathcal{O}(|S|^{c-1})$ vertices, where $c = \max\{|V(H)| : H \in \mathcal{H}\}$ (we assume $c > 1$, as otherwise the problem is trivial). Since the annotated version is in NP (as an unparameterized problem), this compression, together with an algorithm to obtain S, concludes the proof of Theorem 2. Note that we do not require inclusion-wise minimal HDSs of size larger than k to be contained in $E(S)$.

Lemma 2 *Let* $c = \max\{|V(H)| : H \in \mathcal{H}\}$. *There is an algorithm that, given an instance* (G, k) *of* $\mathcal{H}$-FREE EDGE DELETION *and a set* $S \subseteq V(G)$ *such that every inclusion-wise minimal HDS of size at most* k *is contained in* $E(S)$*, outputs in polynomial time a set* U*, where* $S \subseteq U \subseteq V(G)$ *and* $|U| \leq \mathcal{O}(|S|^{c-1})$*, such that* (G, k) *is a yes-instance if and only if* $(G[U], S, k)$ *is a yes-instance of* ANNOTATED $\mathcal{H}$-FREE EDGE DELETION.

Proof Let (G, k) and S be as in the statement, we construct U in the following way. Add all vertices of S to U. For every set M of at most $c - 1$ vertices (possibly empty) in S and every subset $F \subseteq E(M)$, add to U all vertices of up to one subgraph $H \in \mathcal{H}$ induced in $G - F$ such that $V(H) \cap S = M$ (if there is more than one, choose any). Clearly $S \subseteq U \subseteq V(G)$, $|U| \leq |S| + c^2 \cdot 2^{\binom{c-1}{2}} \cdot |S|^{c-1}$, and U can be constructed in polynomial time. We claim that (G, k) is a yes-instance if and only if $(G[U], S, k)$ is a yes-instance of ANNOTATED $\mathcal{H}$-FREE EDGE DELETION.

Suppose first that (G, k) is a yes-instance, that is, there is a set of edges $F \subseteq E(G)$ of size at most k such that $G - F$ is $\mathcal{H}$-free; without loss of generality suppose that F is inclusion-wise minimal. By the claim's assumption, F is contained in $E(S)$. Since $G[U]$ is an induced subgraph of G, $G[U] - F$ is also $\mathcal{H}$-free. Thus $(G[U], S, k)$ is a yes-instance of ANNOTATED $\mathcal{H}$-FREE EDGE DELETION.

Suppose then that $(G[U], S, k)$ is a yes-instance of ANNOTATED $\mathcal{H}$-FREE EDGE DELETION, that is, there is a set of edges $F \subseteq E(S)$ of size at most k such that $G[U] - F$ is $\mathcal{H}$-free. We claim F is an HDS of G, too. Suppose that, to the contrary, $G - F$ has an induced subgraph $H \in \mathcal{H}$. If $V(H) \subseteq S$, then because $S \subseteq U$ we would have that H is an induced subgraph of $G[U] - F$, a contradiction. Let then $M = V(H) \cap S$, and note that $|M| \leq c - 1$. Since $F \subseteq E(S)$, observe that the non-edges of H deleted by F are all in $E(M)$ and hence H is an induced subgraph in $G - (F \cap E(M))$ as well. By the construction of U, there is an induced subgraph $H' \in \mathcal{H}$ in $G - (F \cap E(M))$ such that $V(H') \cap S = M$ and all of the vertices of H' were added to U. Since $F \setminus E(M)$ deletes only edges in $E(S) \setminus E(M)$, H' is an induced subgraph in $G - F$ as well. But all the vertices of H' are in U, so H' is an induced subgraph in $G[U] - F$, a contradiction. This shows F must be an HDS of G and hence (G, k) is a yes-instance of $\mathcal{H}$-FREE EDGE DELETION. □

3 Kernel

In this section, we prove Theorem 2. As discussed below the statement of Theorem 2, this yields the proof of Theorem 1 and thus the kernel. Throughout, let (G, k) be an instance of {CLAW,DIAMOND}-FREE EDGE DELETION.

We first define a simple decomposition of {claw, diamond}-free graphs, which follows from the fact that they are precisely the line graphs of triangle-free graphs, as shown by Metelsky and Tyshkevich [26]. For a {claw, diamond}-free graph G', let $\mathcal{B}(G')$ be the family of vertex sets, called *bags*, containing:

- every maximal clique of G', and
- a singleton $\{v\}$ for each simplicial vertex v of G' (i.e., each vertex whose neighborhood is a clique).

Lemma 3 *Let G' be a* {*claw*, *diamond*}*-free graph. Consider the family $\mathcal{B}(G')$ of bags of G'. Then:*

(a) *every non-isolated vertex of G' is in exactly two bags;*
(b) *for every edge $uv \in E(G')$ there is exactly one bag containing both u and v;*
(c) *every two bags have at most one vertex in common;*
(d) *if two bags A, B have a common vertex v, then there is no edge between $A - v$ and $B - v$.*

Moreover, $|\mathcal{B}(G')| \leq |V(G')| + |E(G')|$ and the family $\mathcal{B}(G')$ can be computed in polynomial time.

Proof From the definitions of Section 3 and Theorem 5.2 of [26] it follows that {claw, diamond}-free graphs are precisely the *linear r-minoes* for $r = 2$, that is, graphs G' such that every vertex belongs to at most two maximal cliques and every edge belongs to exactly one maximal clique. In particular every edge of G' is contained in exactly one bag, which proves (b).

Let v be any non-isolated vertex of G'. If the neighborhood of v is a clique in G', then $N[v]$ is the only maximal clique containing v – hence v is in exactly two bags: the maximal clique and the singleton $\{v\}$, by definition. If the neighborhood of v is not a clique, then v has neighbors a, b that are not adjacent – hence v is contained in at least two bags: the maximal clique containing va and the (different) maximal clique containing vb. As G' is a linear 2-mino, v is not in any other maximal clique. Since v is not simplicial, by the definition of $\mathcal{B}(G')$ we conclude that also in this case v is in exactly two bags. This concludes the proof of (a).

Since all bags induce cliques in G', two different bags cannot have more than one vertex in common, as this would imply that an edge joining them is contained in both of them. This proves (c).

Finally, if two bags A, B had a common vertex v and there was an edge between $a \in A - v$ and $b \in B - v$, then since A is a maximal clique not containing b, there would be a vertex $a' \in A$ non-adjacent to b. But then the vertices a, a', b, v would induce a diamond subgraph in G', a contradiction. This proves (d).

To see that $|\mathcal{B}(G')| \leq |V(G')| + |E(G')|$, note that every bag of $\mathcal{B}(G')$ is either a singleton bag or it contains an edge. The number of singleton bags is bounded by $|V(G')|$, while the number of bags containing an edge is bounded by $|E(G')|$ due to (b). In order to compute $\mathcal{B}(G')$, it suffices to construct first singleton bags for all simplicial and isolated vertices, and then for every edge of G add the unique maximal clique containing it, constructed in a greedy manner. □

Now run the algorithm of Lemma 1 on instance (G, k). In case the algorithm concludes it is a no-instance, we return a trivial no-instance of ANNOTATED {CLAW,DIAMOND}-FREE EDGE DELETION as the output of the compression. Otherwise, let X be the obtained modulator; that is, X is a subset of $V(G)$ of size at most $4k$ such that every induced claw and diamond in G has an edge in $E(X)$. In particular, $G - X$ is a {claw, diamond}-free graph, so using Lemma 3 we compute in polynomial time the family of bags $\mathcal{B}(G - X)$. When referring to bags, we will refer to $\mathcal{B}(G - X)$ only, and implicitly use Lemma 3 to identify, for each non-isolated vertex v in $G - X$, *the two bags containing* v, and for each edge e of $G - X$, *the bag containing* e.

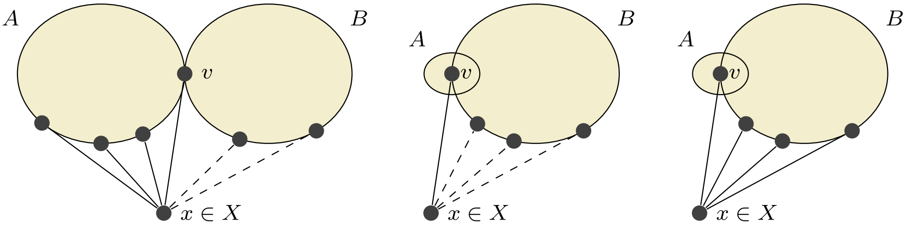
Knowing the structure of $G - X$, we proceed by describing the adjacencies between X and $G - X$. The following definition will play a central role. For $x \in X$, we call a bag B of $G - X$ *attached* to x if:

- B is fully adjacent to x, and
- if $B = \{v\}$ for some vertex v which is not isolated in $G - X$, then the other bag containing v is not fully adjacent to x.

We call a bag *attached* if it is attached to some $x \in X$. The next two propositions show that adjacencies between X and $G - X$ are fully determined by the attachment relation, see Fig. 1.

Lemma 4 *Let $B \in \mathcal{B}(G - X)$ be a bag such that some vertex $x \in X$ has at least two neighbors in B. Then B is attached to x.*

Proof Suppose x is adjacent to $u, v \in B$. If x was non-adjacent to some vertex $w \in B$, then since B induces a clique, the vertices x, u, v, w would induce a diamond subgraph in G (Fig. 2a). However, no edge of this induced diamond would be in $E(X)$, contradicting the properties of X as a modulator. Therefore, all vertices of B are adjacent to x (and $|B| > 1$), so B is attached to x. □



(a) x attached to A, not to B

(b) x attached to a singleton bag

(c) x attached to B, not to A

Fig. 1 Possible ways in which a vertex in X can neighbor a vertex v in $G - X$ and the two bags containing it

Lemma 5 *Let v be a vertex in $G - X$ adjacent to a vertex $x \in X$. Then there is exactly one bag in $\mathcal{B}(G - X)$ that contains v and is attached to x.*

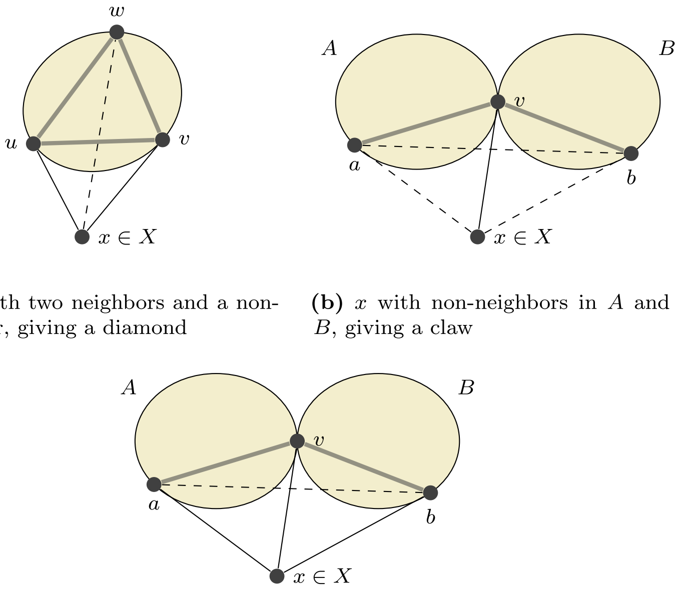
Proof If v is an isolated vertex in $G - X$, then $\{v\}$ is the only bag containing v and is by definition attached to x.

Otherwise, let A, B be the two bags containing v. If one of the bags is a singleton, say $A = \{v\}$, then B, being unequal to A, contains some other vertices. If at least one vertex of $B \setminus \{v\}$ is adjacent to x, then it follows from Lemma 4 that B is attached to x and A is not. Otherwise, i.e. if no vertices of $B \setminus \{v\}$ are adjacent to x, then by definition A is attached to x and B is not.

It remains to consider the case when both $A - v$ and $B - v$ are not empty; see Fig. 2b and c. Suppose that x is adjacent to a vertex $a \in A - v$ and a vertex $b \in B - v$. Then a, b are non-adjacent by Lemma 3(d), so vertices v, a, b, x induce a diamond subgraph in G. However, no edge of this diamond is in $E(X)$, a contradiction.

Suppose x is non-adjacent to a vertex $a \in A - v$ and a vertex $b \in B - v$. Then a, b are non-adjacent by Lemma 3(d), so vertices x, a, b, v induce a claw subgraph in G. However, no edge of this claw is in $E(X)$, again a contradiction.

Therefore, if x is adjacent to a vertex in $A - v$, then A is attached to x (by Lemma 4) and x must be non-adjacent to all of $B - v$, implying B is not attached to x. Otherwise, if x is non-adjacent to all vertices in $A - v$, then x must be adjacent to every vertex of $B - v$. This means B is attached to x and A is not. □



(a) x with two neighbors and a non-neighbor, giving a diamond

(b) x with non-neighbors in A and B, giving a claw

(c) x with neighbors in A, B, giving a diamond

Fig. 2 Adjacencies between X and $G - X$ that lead to a contradiction

We can now limit the number of attached bags by $2|X|$, which is linear in k.

Lemma 6 *For any $x \in X$, there are at most two bags in $\mathcal{B}(G - X)$ attached to x.*

Proof Let $x \in X$. We first show that bags attached to x must be pairwise disjoint and non-adjacent. If two bags attached to x contained a common vertex v, then v would be adjacent to x and, by Lemma 5, at most one of the bags would be attached to x, a contradiction.

If there was an edge uv between two different bags attached to x, then its endpoints u and v would be adjacent to x and, by Lemma 4, the bag containing the edge uv would be attached to x. But we have just shown that bags attached to x are disjoint, so no other bag attached to x could contain u or v, a contradiction. Therefore, every two bags attached to x are disjoint and non-adjacent.

Suppose that there are three or more bags adjacent to x. Let u, v, w be any vertices contained in three different bags. By the above observations, u, v, w are pairwise different and non-adjacent. Hence, vertices x, u, v, w induce a claw in G that has no edges in $E(X)$, a contradiction. □

Having limited the number of attached bags, we want to show that unattached bags intersect solutions only in a simple way. The following technical proposition will help handle cases involving diamonds.

Lemma 7 *Let H be a subgraph (not necessarily induced) of G isomorphic to a diamond. Let $B \in \mathcal{B}(G - X)$ be an unattached bag containing at least two vertices of H. Then B contains all vertices of H.*

Proof Let u, v be two vertices of H in B. Let w be a vertex of H adjacent to u and v in H (note that since H is a diamond, there always is such a vertex). Then w is also adjacent to u and v in G. Vertex w cannot be in X, as otherwise Lemma 4 would contradict the assumption that B is unattached. Hence, w is in $G - X$. Let A be the bag containing the edge uw. If w was not in B, then $B \neq A$ and vw would be an edge going between $v \in B - u$ and $w \in A - u$, contradicting Lemma 3(d). Therefore, $w \in B$.

Repeating this argument for the fourth vertex of the diamond H and an appropriate pair of vertices from $\{u, v, w\}$, all the vertices of H can be shown to be in B. □

It turns out that one may need to delete an edge of an unattached bag B, but in this case the intersection of any minimal HDS F with the edges of B has a very special structure: deleting the edges of F makes some of the vertices of B isolated, whereas the rest of B remains a smaller clique. This will later allow us to take only a limited number of unattached bags into account.

Lemma 8 *Let F be a minimal HDS of G and let $B \in \mathcal{B}(G - X)$ be an unattached bag. Then $G[B] - F$ consists of a clique and a number of isolated vertices.*

Proof Let $B' \subseteq B$ be the set of vertices that are not isolated in $G[B] - F$. Consider the set $F' = F \setminus E(B')$. The graph $G - F'$ is obtained from $G - F$ by adding back all edges between vertices in B'. Thus the bag B induces in $G - F'$ a clique on B' plus isolated vertices $B \setminus B'$. We claim that F' is an HDS. By the minimality of F, this will imply that $F = F'$ and hence the claim.

Suppose to the contrary that $G - F'$ contains an induced claw or diamond H. Since $G - F$ contains neither an induced claw nor a diamond, H has an edge e in $F \cap E(B')$.

If H is a diamond in $G - F'$, then since e has both endpoints in B, by Lemma 7 we infer that all vertices of H are in B. But this contradicts that B induces a clique plus isolated vertices in $G - F'$.

If H is a claw in $G - F'$, then let c be its center and v, u_1, u_2 its leaves, so that $e = cv$. Since $e \in E(B')$, its endpoint c is in B', meaning c is not isolated in $G[B] - F$. Let w be a neighbor of c in $G[B] - F$. We show that vertices c, w, u_1, u_2 induce a claw in $G - F$. Consider where the leaves u_i may be. If $u_i \in B$ (for $i = 1$ or 2), then vertices c, v, u_i induce two legs of a claw (a P_3) in $G[B] - F'$, contradicting that $G[B] - F'$ is a clique plus isolated vertices. Note that $u_i \notin B$ implies that $u_i \neq w$. If $u_i \in X$, then since u_i is adjacent to $c \in B$ and B is not attached, by Lemma 4 we have that u_i cannot be adjacent to $w \in B$ in G. If $u_i \in G - (B \cup X)$, then it is in the bag A containing the edge cu_i and, by Lemma 3, $u_i \in A - c$ is not adjacent to $w \in B - c$ in G. In either case $u_1 w$ and $u_2 w$ are non-edges in G, thus also in $G - F$. By assumption, $u_1 u_2$ is a non-edge in $G - F'$, thus also in $G - F$. We showed that $u_i \notin B$, so $cu_i \in E(G - F')$ are also edges in $G - F$. Finally, $cw \in E(G[B]) \setminus F$, so indeed the vertices c, w, u_1, u_2 induce a claw in $G - F$, a contradiction. □

To obtain a compressed description of the problem, one ingredient remains: limiting the size of bags that may need deletions.

Lemma 9 *If K is a clique in G with at least $2k + 2$ vertices, then every HDS F of G of size at most k satisfies $F \cap E(K) = \emptyset$.*

Proof By contradiction, assume there exists $uv \in F$ with $u, v \in K$. However, then for every two distinct $w_1, w_2 \in K \setminus \{u, v\}$, the subgraph induced in $G - uv$ by u, v, w_1, w_2 is a diamond. As $|K| \geq 2k + 2$, we can find k edge-disjoint diamonds formed in this way in $G - uv$. Consequently, F needs to contains at least k edges apart from uv, a contradiction. □

Corollary 2 *Let $B \in \mathcal{B}(G - X)$ be a bag with at least $2k + 2$ elements. Then for every HDS F of G of size at most k, $F \cap E(B) = \emptyset$. If furthermore B is attached to $x \in X$, then $F \cap E(B \cup \{x\}) = \emptyset$.*

Proof Follows directly from Lemma 9, since every bag B is a clique, if B is attached to $x \in X$, then $B \cup \{x\}$ is a clique as well. □

We are ready to present the main step of the compression procedure for {CLAW,DIAMOND}-FREE EDGE DELETION.

Lemma 10 *One can in polynomial time find a set $S \subseteq V(G)$ of size $\mathcal{O}(k^4)$ such that every minimal HDS of size at most k is contained in $E(S)$.*

Proof Call a bag *small* if it has less than $2k+2$ vertices, *big* otherwise. We *mark* the following bags:

- every small attached bag,
- every small unattached bag that shares a vertex with some small attached bag,
- for every vertex pair $x, y \in X$, we mark up to $k+1$ small unattached bags of size at least two that have a vertex in $N(x) \cap N(y)$ (if there are more such bags, we mark any $k+1$ of them).

Let S be the set of all vertices in marked bags and in X. Let us first show that $|S| = \mathcal{O}(k^4)$. By the construction of X in Lemma 1, we have that $|X| \leq 4k$. By Lemma 6, there are at most $2|X|$ attached bags. Hence, there are at most $2|X| \cdot (2k+1)$ vertices in small attached bags. Since each vertex of $G - X$ is in at most two bags, there are at most $2|X| \cdot (2k+1)$ small unattached bags that share a vertex with small attached bags. In the final point we mark at most $|X|^2 \cdot (k+1)$ small bags. Therefore, we mark at most $2|X| + 2|X| \cdot (2k+1) + |X|^2 \cdot (k+1) = \mathcal{O}(k^3)$ small bags in total. The set $S \setminus X$ contains at most $(2k+1)$ times as many vertices in total, which together with $|X| \leq 4k$ implies that $|S| = \mathcal{O}(k^4)$.

We want to show that a minimal HDS never deletes any edges in unmarked bags. Let Z be the set of edges that are either contained in a marked bag, or in $E(X)$, or connect a vertex of a marked bag with a vertex of X. Note that $Z \subseteq E(S)$, but the inclusion may be strict, due to an edge going between two vertices of some marked bags that belongs to an unmarked bag. Let F be a minimal HDS of size at most k. We will show that $F' = F \cap Z$ is also an HDS. By the minimality of F, this will imply that $F = F' \subseteq Z \subseteq E(S)$, and hence the proof of the lemma will be concluded. □

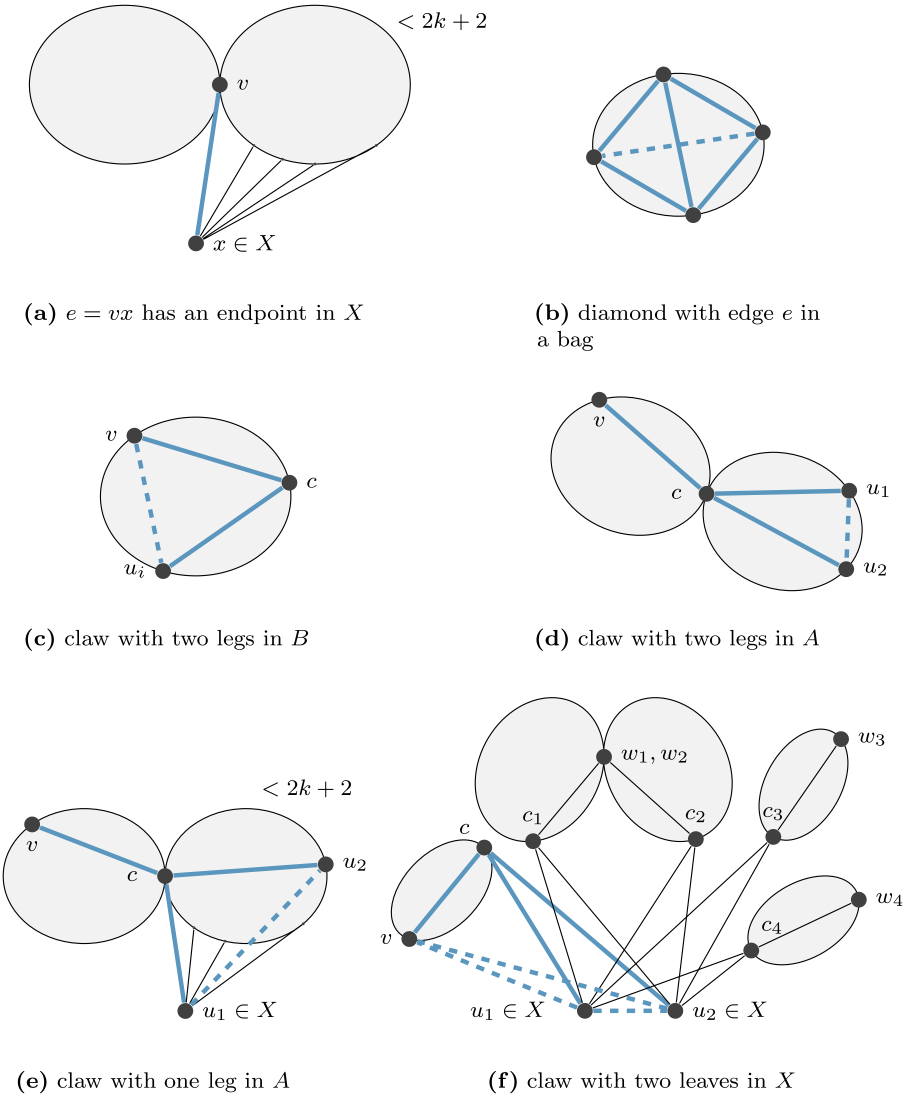
Claim 1 If a bag does not induce a clique plus isolated vertices in $G - F'$, then it is a small attached bag.

Proof First consider $G - F$. By Lemma 8, every unattached bag induces a clique plus isolated vertices in $G - F$. By Corollary 2, every big bag induces a clique in $G - F$. Hence, if a bag does not induce a clique plus isolated vertices in $G - F$, then it is a small attached bag. Suppose now that a bag does not induce a clique plus isolated vertices in $G - F'$. Then it necessarily contains an edge of $F' \subseteq Z$ and thus must be marked. We infer that this bag induces the same subgraph in $G - F$ as in $G - F'$. Therefore, it must be small and attached.

Suppose to the contrary that $G - F'$ contains an induced claw or diamond H. Since $G - F$ contained none, H must have an edge $e \in F \setminus F' = F \setminus Z$. We consider the following cases depending on the location of e, each leading to a contradiction; see Fig. 3.

Case 1: edge e has an endpoint in the modulator X.

Then $e = vx$ for some $x \in X$ and $v \in V(G)$. If $v \in X$, then $e \in E(X) \subseteq Z$, contradicting $e \in F \setminus Z$. Otherwise, by Lemma 5, there is a bag B containing v that is attached to x. Since $e \in F$, by Corollary 2 we infer that B has less than



(a) $e = vx$ has an endpoint in X

(b) diamond with edge e in a bag

(c) claw with two legs in B

(d) claw with two legs in A

(e) claw with one leg in A

(f) claw with two leaves in X

Fig. 3 The different situations where a claw or diamond (*thick blue* edges and *dashed* non-edges) might appear in $G - F'$, each leading to a contradiction

$2k + 2$ elements. But then B is a small, attached, and hence marked bag, implying $e \in Z$, a contradiction.

Case 2: edge e has both endpoints in $G - X$ (and thus e is in $G - X$).

Let B be the bag containing e. Since $e \in F$, B is a small bag by Corollary 2. Since $e \notin Z$, B is not a marked bag. Since small attached bags are marked, B is unattached. By Claim 1, B induces a clique plus isolated vertices in $G - F'$.

Case 2a: H is a diamond (in $G - F'$).

Then the endpoints of e are in B, hence by Lemma 7 all vertices of H are in B. But B induces a clique plus isolated vertices in $G - F'$, a contradiction.

Case 2b: H is a claw (in $G - F'$).

Let c be the center of the claw H and let v, u_1, u_2 be its leaves, so that $e = cv$. Let A be the other bag containing c.

If u_i was in B (for $i = 1$ or 2), then B would not induce a clique plus isolated vertices in $G - F'$ because u_i, c, v induces a P_3, a contradiction.

If $u_i \notin X$, then u_i is in the bag containing cu_i but not in B, which means that u_i is in A. If both u_1, u_2 were not in X, then A would not induce a clique plus isolated vertices in $G - F'$ (because u_1, c, u_2 induces a P_3). By Claim 1, A would be a small attached bag that shares the vertex c with B, implying that B is marked, a contradiction.

If exactly one leaf of the claw is in X, e.g., $u_1 \in X$ and $u_2 \in G - X$, then u_2 is in A (as above). Because c is adjacent to $u_1 \in X$, by Lemma 5 we infer that one of A, B is attached to u_1. Since B is unattached, A is attached to u_1, so u_1u_2 is an edge in G. Since u_1u_2 is not an edge in $G - F'$, we have that $u_1u_2 \in F' \subseteq F$. By Corollary 2 we infer that A is a small bag. It is also attached, and therefore B is marked, again a contradiction.

If both u_1, u_2 are in X, then note that B is an unattached bag of size at least two that has a vertex (namely c) in the common neighborhood of u_1 and u_2. By the definition of marked bags and as B was not marked in the third point, at least $k + 1$ different marked bags $B_1, \ldots, B_{k+1}$ are unattached, have size at least two, and have some vertex, respectively $c_1, c_2, \ldots, c_{k+1}$, in the common neighborhood of u_1 and u_2. If $c_i = c_j$ for some i, j with $1 \leq i < j \leq k + 1$, then B_i, B_j are the two bags that contain c_i. Since c_i is adjacent to u_1, one of those bags is attached to u_1 by Lemma 5, a contradiction. Hence, $c_i \neq c_j$ for all $1 \leq i < j \leq k + 1$. Let w_i be any vertex different from c_i in B_i. Since B_i is unattached, w_i is non-adjacent to u_1 and u_2 in G by Lemma 4. Clearly, c_i is adjacent to w_i, u_1, u_2 in G. Therefore, vertices c_i, w_i, u_1, u_2 induce $k + 1$ edge-disjoint claws in $G - u_1u_2$. Since u_1, u_2 are leaves of the claw H in $G - F'$, they are non-adjacent in $G - F$. Hence, for each i with $1 \leq i \leq k + 1$, one of the edges c_iw_i, c_iu_1, c_iu_2 must be deleted by F. But $|F| \leq k$, a contradiction. □

We can now conclude the proof of Theorem 2: given an instance (G, k), we first apply Lemma 10 to obtain a set $S \subseteq V(G)$, then apply Lemma 2 to the set S obtaining a set $S \subseteq U \subseteq V(G)$, and return an instance $(G[U], S, k)$ of Annotated $\mathcal{H}$-free Edge Deletion. The correctness of this procedure follows from Lemmata 10 and 2.

Note that $|S| = \mathcal{O}(k^4)$, thus $|U| = \mathcal{O}(k^{12})$ and the adjacency matrix of the output instance can be encoded with $\mathcal{O}(k^{24})$ bits.

4 Hardness

In this section, we prove Theorem 3, which states that the problems we consider cannot be solved in subexponential time, under the Exponential Time Hypothesis (ETH). Let us recall that this hypothesis, formulated by Impagliazzo, Paturi and Zane [22], states that there exists a positive real number s such that 3SAT with n variables cannot be solved in time $\mathcal{O}(2^{sn})$. The Sparsification Lemma of [22] allows to strengthen this assumption to functions subexponential in the size of the formula (the number variables n plus the number of clauses m of the input formula), and not just the number of variables. More precisely, unless ETH fails, 3SAT cannot be solved in time $\mathcal{O}(2^{s(n+m)})$ for some $s > 0$. In Theorem 3, we give a reduction that, given a 3SAT instance ϕ, outputs in polynomial time an equivalent instance (G, k) of {CLAW,DIAMOND}-FREE EDGE DELETION where k (the number of allowed deletions) is linear in the size of ϕ. Composing this reduction with any subexponential parameterized algorithm for the problem would imply a subexponential algorithm for 3SAT, contradicting ETH; this shows how Theorem 3 implies Corollary 1.

Our approach to proving Theorem 3 is to consider CLAW-FREE EDGE DELETION in graphs where diamonds are not present and cannot appear after any edge deletions. That is, we shall actually prove the following result.

Theorem 4 *There exists a polynomial-time reduction that, given an instance ϕ of* 3SAT *with n variables and m clauses, outputs an instance (G, k) of* CLAW-FREE EDGE DELETION *such that (a) (G, k) is a yes-instance if and only if ϕ is satisfiable, (b) $|V(G)|, k = \mathcal{O}(n + m)$, (c) G is $\{K_4$, diamond$\}$-free, and (d) $\Delta(G) = 6$.*

Theorem 3 follows, since an instance of CLAW-FREE EDGE DELETION with no diamond or K_4 as an induced subgraph is a yes-instance if and only if it is a yes-instance of {CLAW,DIAMOND}-FREE EDGE DELETION (the solution sets are even identical, because deleting an edge from a $\{K_4$, diamond$\}$-free graph cannot create a diamond). Note that since $\Delta(G) = 6$, both in Theorem 4 and in Theorem 3, we have that $|E(G)| \leq 3|V(G)|$ and under ETH there is even no subexponential-time algorithm in terms of the number of edges of the graph. We remark that the original NP-hardness reduction for CLAW-FREE EDGE DELETION of Yannakakis [28][2] actually implies that this problem cannot be solved in subexponential parameterized time; however, the constructed graph contains a lot of diamonds and the reduction cannot be easily adapted to our setting.

The remainder of this section is devoted to the proof of Theorem 4.

Let ϕ be an instance of 3SAT – a formula in conjunctive normal form whose every clause has exactly three literals with three different variables (formulas with clauses

[2] Yannakakis [28] proves NP-hardness of LINE GRAPH EDGE DELETION, but the same reduction works also for CLAW-FREE EDGE DELETION.

of at most three, possibly equal literals can easily be transformed to this form via standard reductions; see e.g. [18]). Let $\mathcal{V}(\phi)$ be the set of variables of ϕ; write $c \in \phi$ for clauses c of ϕ and write $x \in \mathcal{V}(c)$ for the three variables occurring in the clause. We write $\perp, \top$ for the `false`,`true` values assigned to the variables, respectively.

Let us begin by defining the clause gadget for each clause $c \in \phi$. We construct the following graph G_c (see Fig. 4). G_c has 19 vertices: a central vertex u_c and vertices v_c^x, $\tilde{v}_c^x$, w_c^x, $\tilde{w}_c^x$, t_c^x, $\tilde{t}_c^x$ for each $x \in \mathcal{V}(c)$; and 27 edges arranged in 9 triangles: $\{u_c, v_c^x, \tilde{v}_c^x\}$, $\{v_c^x, w_c^x, \tilde{w}_c^x\}$, $\{v_c^x, t_c^x, \tilde{t}_c^x\}$ for each $x \in \mathcal{V}(c)$.

Not satisfying a literal of variable $x \in \mathcal{V}(c)$ will correspond to deleting the edge $t_c^x \tilde{t}_c^x$ (thick blue edges on Fig. 4). We show that in a minimum HDS at least one thick edge shall not be deleted.

Claim 2 Let F be an HDS of G_c. Then F deletes at least 7 edges in $E(G_c) \setminus \{t_c^x \tilde{t}_c^x \mid x \in \mathcal{V}(c)\}$. Furthermore, if it deletes exactly 7 edges in this set, then F does not delete $t_c^x \tilde{t}_c^x$ for some $x \in \mathcal{V}(c)$.

Proof We drop the subscript c of all vertices for clarity. To see that F deletes at least 7 edges in $E(G_c) \setminus \{t^x \tilde{t}^x \mid x \in \mathcal{V}(c)\}$ it suffices to notice that this set contains all edges of 7 edge-disjoint induced claw subgraphs of G_c: $\{u, \tilde{v}^x \mid x \in \mathcal{V}(c)\}$ and $\{v^x, u, w^x, t^x\}$, $\{v^x, \tilde{v}^x, \tilde{w}^x, \tilde{t}^x\}$ for $x \in \mathcal{V}(c)$.

Suppose now that F deletes exactly 7 edges in the above set and deletes all edges $t^x \tilde{t}^x$ for $x \in \mathcal{V}(c)$. Then it must delete exactly one edge of each of the 7 claws above. In particular F deletes exactly two edges incident to v^x for every $x \in \mathcal{V}(c)$. A direct

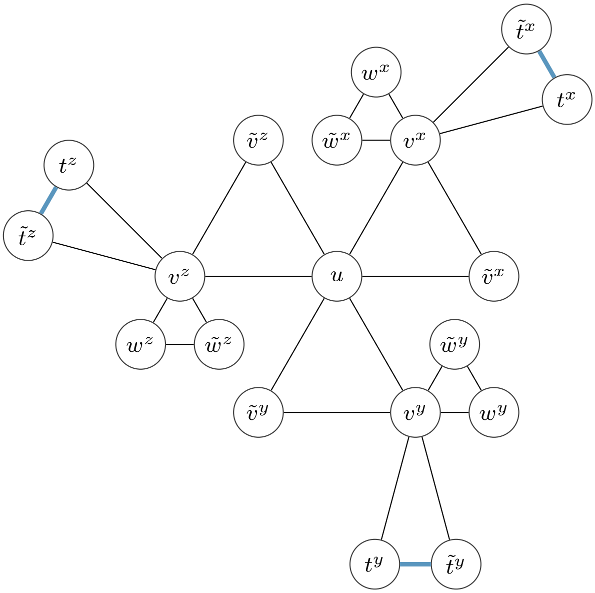


Fig. 4 The clause gadget G_c used in the reduction

check shows that those two edges are $v^x t^x$ and $v^x \tilde{t}^x$, as otherwise there would be a claw left centered at v^x (recall that we assumed the edge $t^x \tilde{t}^x$ to be deleted). But as $|F \setminus \{t^x \tilde{t}^x \mid x \in \mathcal{V}(c)\}| = 7$, it follows that F deletes at most one edge incident to u, leaving a claw with center u in $G_c - F$: the one with leaves $\{v^x \mid x \in \mathcal{V}(c)\}$ or the one with leaves $\{\tilde{v}^x \mid x \in \mathcal{V}(c)\}$. □

Let us now define the variable gadget (see Fig. 5), as a part of the final graph G constructed for ϕ. Take G to be the disjoint union of G_c for all clauses c of ϕ. For each variable x of ϕ, add four vertices $t^x_\top, t^x_\bot, s^x_\top, s^x_\bot$ to G and then for each t-vertex (with tilde or not) in a clause gadget add a corresponding s-vertex to G. That is,

$$\begin{aligned} V(G) = &\{u_c \mid c \in \phi\} \cup \{v^x_c, \tilde{v}^x_c, w^x_c, \tilde{w}^x_c, t^x_c, \tilde{t}^x_c, s^x_c, \tilde{s}^x_c \mid x \in \mathcal{V}(c), c \in \phi\} \\ &\cup \{t^x_\top, t^x_\bot, s^x_\top, s^x_\bot \mid x \in \mathcal{V}(\phi)\}. \end{aligned}$$

Let $c_1, c_2, \ldots, c_{p(x)}$ be the clauses in which x occurs positively, and let $d_1, d_2, \ldots, d_{q(x)}$ be the clauses in which x occurs negatively. To the edges defined before (between vertices with the same subscript c) we add the following ones (between vertices with the same superscript x): an edge between every t-vertex and its s-counterpart and an edge between every two consecutive vertices in the following cycle of length $1 + 2p(x) + 1 + 2q(x)$:

$$\begin{aligned} &t^x_\top, \quad t^x_{c_1}, \tilde{t}^x_{c_1}, \; t^x_{c_2}, \tilde{t}^x_{c_2}, \; \ldots, \; t^x_{c_{p(x)}}, \tilde{t}^x_{c_{p(x)}}, \\ &t^x_\bot, \quad t^x_{d_1}, \tilde{t}^x_{d_1}, \; t^x_{d_2}, \tilde{t}^x_{d_2}, \; \ldots, \; t^x_{d_{q(x)}}, \tilde{t}^x_{d_{q(x)}}, \quad t^x_\top. \end{aligned}$$

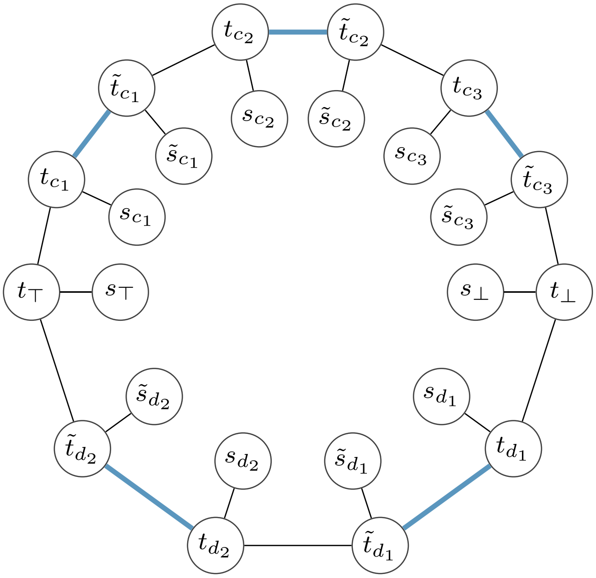


Fig. 5 The variable gadget G^x used in the reduction

The variable gadget G^x is the subgraph of G induced by the $2(1 + 2p(x) + 1 + 2q(x))$ vertices named t or s with superscript x (so G^x is a cycle on the t-vertices, each with a pendant s-vertex attached).

Define $E^x_\perp, E^x_\top$ to be the set of even and odd edges on the above cycle, respectively (so that for all i, $t^x_{c_i}\tilde{t}^x_{c_i} \in E^x_\perp$ and $t^x_{d_i}\tilde{t}^x_{d_i} \in E^x_\top$). The crucial property is that for any clause c containing x, assigning $x \mapsto b$ satisfies the clause c if and only if $t^x_c\tilde{t}^x_c$ is *not* in E^x_b. We show that in a minimum HDS exactly one of $E^x_\perp, E^x_\top$ is deleted.

Claim 3 Let $F \subseteq E(G^x)$ be an HDS of G^x. Then F deletes at least $p(x)+q(x)+1$ edges of G^x. Furthermore, if it deletes exactly that many, then either $F = E^x_\perp$ or $F = E^x_\top$.

Proof To show that F deletes at least $p(x)+q(x)+1$ edges of G^x it suffices to notice that each of the $2p(x) + 2q(x) + 2$ vertices of the cycle is a center of a different claw and each edge deletion hits at most two such claws.

If F deletes exactly that many edges in G^x, then each edge of F must be contained in exactly two such claws (hence only edges of the cycle get deleted), and no claw may be hit by two edges (hence no two incident edges get deleted). This means either exactly the even edges $E^x_\perp$ or exactly the odd edges $E^x_\top$ of the cycle are deleted by F. □

We let $k = 7m + \sum_{x\in\mathcal{V}(\phi)}(p(x) + q(x) + 1) = 7m + 3m + n$, and it is straightforward to verify that G is $\{K_4, \text{diamond}\}$-free and $\Delta(G) = 6$. Thus, the following lemma encapsulates the final check needed to conclude the proof of Theorem 4.

Lemma 11 *G has an HDS of size at most k if and only if ϕ is satisfiable.*

Proof (Left to Right) Since G_c, G^x are induced subgraphs of G and the edge sets $E(G^x)$ for $x \in \mathcal{V}(\phi)$ and $E(G_c) \setminus \{t^x_c\tilde{t}^x_c \mid x \in \mathcal{V}(c)\}$ for $c \in \phi$ are pairwise disjoint, Claims 2 and 3 imply than any HDS of G has size at least k. Furthermore, equality holds only if it holds in both claims. Let F be an HDS of size equal to k. From Claim 3, we infer that $F \cap E(G^x) = E^x_{b(x)}$ for some $b : \mathcal{V}(\phi) \to \{\perp, \top\}$. From Claim 2, for each clause c of ϕ, there is a variable x in c such that $t^x_c\tilde{t}^x_c$ is not deleted. This means $F \cap E(G^x) = E^x_{b(x)}$ does not contain this edge, so by construction we infer that assigning $x \mapsto b(x)$ satisfies clause c. Therefore, each clause is satisfied by assignment b.

(Right to Left) Let $b : \mathcal{V}(\phi) \to \{\perp, \top\}$ be a satisfying assignment for ϕ. We show that by deleting $\bigcup_{x\in\mathcal{V}(\phi)} E^x_{b(x)}$ and 7 more edges in each clause gadget we can get a claw-free graph. For each clause $c \in \phi$, there is a variable y such that assigning $y \mapsto b(y)$ satisfies c, which by the construction means that $t^y_c\tilde{t}^y_c \notin E^y_{b(y)}$. Let $F_c = \{u_c v^y_c, u_c\tilde{v}^y_c, v^y_c\tilde{v}^y_c\} \cup \{v^x_c t^x_c, v^x_c\tilde{t}^x_c \mid x \in \mathcal{V}(c), x \neq y\}$. We claim the set $F := \bigcup_{x\in\mathcal{V}(\phi)} E^x_{b(x)} \cup \bigcup_{c\in\phi} F_c$ is an HDS of G; note that we have that $|F| = k$.

Observe that if $t_c^x\tilde{t}_c^x$ gets deleted by F for some $c \in \phi, x \in \mathcal{V}(c)$, then $t_c^x\tilde{t}_c^x \in E_{b(x)}^x$. Hence, by the construction we infer that assigning $x \mapsto b(x)$ does not satisfy c, which by the definition of F_c implies that edges $v_c^x t_c^x$, $v_c^x\tilde{t}_c^x$ do not get deleted by F.

Suppose now that $G - F$ contains a claw. The center of this claw has degree at least 3 in G, so it is a u-, v-, t-, or $\tilde{t}$-vertex. It cannot be a u-vertex, since the closed neighborhood of u_c for any $c \in \phi$ forms three triangles in G, exactly one of whose edges are deleted, completely, by F_c. So the closed neighborhood of u_c in $G - F$ forms two triangles joined at u_c. The center also cannot be a v-vertex, since the closed neighborhood of v_c^x for any $c \in \phi, x \in \mathcal{V}(c)$ forms three triangles $\{v_c^x, u_c, \tilde{v}_c^x\}$, $\{v_c^x, w_c^x, \tilde{w}_c^x\}$ and $\{v_c^x, t_c^x, \tilde{t}_c^x\}$ in G, for which either the first one gets completely deleted by F_c, or $v_c^x t_c^x$ and $v_c^x\tilde{t}_c^x$ get deleted by F_c. Additionally, $t_c^x\tilde{t}_c^x$ may be deleted by $E_{b(x)}^x$, but by the observation of the previous paragraph, this only occurs in the second case (i.e., when $v_c^x t_c^x$ and $v_c^x\tilde{t}_c^x$ get deleted by F_c). So the closed neighborhood of v_c^x in $G - F$ again forms two triangles joined at v_c^x. Finally, the center cannot be $t_\perp^x$ or $t_\top^x$ for some $x \in \mathcal{V}(\phi)$, since these vertices have degree 2 in $G - F$.

Hence, the center of the claw in $G - F$ must be t_c^x or $\tilde{t}_c^x$ for some $c \in \phi, x \in \mathcal{V}(c)$. Suppose without loss of generality that it is t_c^x. There are four neighbors of t_c^x in G: s_c^x, $\tilde{t}_c^x$, v_c^x, and either $\tilde{t}_d^x$ for some $d \in \phi$ or t_r^x for $r \in \{\perp, \top\}$. Either the edge to $\tilde{t}_c^x$ or the edge to $\tilde{t}_d^x$ (resp. t_r^x) gets deleted by $E_{b(x)}^x$, so for three edges to remain, no others can be deleted. In particular, the edge to v_c^x cannot be deleted and, by the definition of F, this occurs only if $v_c^x\tilde{t}_c^x$ is also not deleted. By the above observation, the edge to $\tilde{t}_c^x$ cannot be deleted. Hence, only $t_c^x\tilde{t}_d^x$ (resp. $t_c^x t_r^x$) gets deleted by F in the neighborhood of t_c^x in G, so this neighborhood forms in $G - F$ a triangle with a pendant vertex. We have obtained a contradiction in all the cases, so F is indeed an HDS of G. □

5 Conclusions

In this paper we have charted the parameterized and kernelization complexity of {CLAW,DIAMOND}-FREE EDGE DELETION by proving that (i) the problem admits a polynomial kernel, and (ii) the simple $5^k \cdot n^{\mathcal{O}(1)}$ branching algorithm following from the observation of Cai [5] cannot be improved to a subexponential parameterized algorithm, unless the ETH fails.

It should not be a surprise for the reader that the results of this paper were obtained while working on kernelization for CLAW-FREE EDGE DELETION. In this problem, by applying the same vertex modulator principle we arrive at the situation where we have a modulator $X \subseteq V(G)$ with $|X| \leq 4k$, and $G - X$ is a claw-free graph. Then, one can use the structural theorem of Chudnovsky and Seymour [8, 9] (see also variants suited for algorithmic applications, e.g., [21]) to understand the structure of $G - X$ and of the adjacencies between X and $G - X$. In essence, the structural theorem yields a decomposition of $G - X$ into *strips*, where each strip induces a graph from one of several basic graph classes; each strip has at most two distinguished cliques (possibly equal) called *ends*, and strips are joined together by creating full adjacencies between disjoint sets of ends. Thus, the whole decomposition looks like a line graph, where every vertex is replaced by a possibly larger strip; indeed, the degenerate case where all the strips are single vertices exactly corresponds to the case of line graphs. As far as base classes are concerned, probably the ones most important for understanding the whole decomposition are proper interval graphs and graphs with independent sets of size at most 2 or 3, in particular, co-bipartite graphs. Thus, we believe that for the sake of showing a polynomial kernel for CLAW-FREE EDGE DELETION, one needs to understand the three special cases when $G - X$ is (a) a line graph, (b) a proper interval graph, and (c) a co-bipartite graph.

We believe that the results of this paper present a progress towards this goal by providing a toolbox useful for tackling case (a). In our proof we have used in several places the fact that we exclude also diamonds. However, much of the structural analysis can translated also to the case when only claws are forbidden, so we hope that similar ideas can be also used for understanding case (a), and consequently how the whole decomposition structure should be dealt with in a polynomial kernel for CLAW-FREE EDGE DELETION. Unfortunately, we are currently unable to make any significant progress in cases (b) and (c), of which case (c) seems particularly difficult.

From another perspective, our positive result gives high hopes for the existence of a polynomial kernel for LINE GRAPH EDGE DELETION, which seems much closer to the topic of this work than CLAW-FREE EDGE DELETION. The problem is that {claw, diamond}-free graphs, or equivalently line graphs of triangle-free graphs, have much nicer structural properties than general line graphs. These properties, encapsulated in Lemma 3, were used several times to simplify the analysis, which would become much more complicated in the case of general line graphs. Also, note that in this paper the considered graph class can be characterized using only two relatively simple forbidden induced subgraphs. In the case of general line graphs, the classic characterization via forbidden induced subgraphs of Beineke [1] involves 9 different obstacles with up to 6 vertices.

Acknowledgment Open access funding provided by Max-Planck Institut für Informatik, Saarbrücken, Germany. We thank the anonymous reviewers for their helpful remarks and comments. Also, we would like to acknowledge the support and extremely productive atmosphere at Dagstuhl Seminar 14451.

References

1. Beineke, L.W.: Characterizations of derived graphs. J. Comb. Theory **9**(2), 129–135 (1970)
2. Bliznets, I., Fomin, F.V., Pilipczuk, M., Pilipczuk, M.: A subexponential parameterized algorithm for proper interval completion. SIAM J. Discrete Math. **29**(4), 1961–1987 (2015). doi:10.1137/140988565
3. Bliznets, I., Fomin, F.V., Pilipczuk, M., Pilipczuk, M.: Subexponential parameterized algorithm for interval completion. In: Krauthgamer, R. (ed.) Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, pp. 1116–1131. SIAM, Arlington (2016). doi:10.1137/1.9781611974331.ch78

4. Brügmann, D., Komusiewicz, C., Moser, H.: On generating triangle-free graphs. Electron. Notes Discrete Math. **32**, 51–58 (2009). doi:10.1016/j.endm.2009.02.008
5. Cai, L.: Fixed-parameter tractability of graph modification problems for hereditary properties. Inf. Process. Lett. **58**(4), 171–176 (1996)
6. Cai, L., Cai, Y.: Incompressibility of H-free edge modification problems. Algorithmica **71**(3), 731–757 (2015). doi:10.1007/s00453-014-9937-x
7. Cai, Y.: Polynomial kernelisation of H-free edge modification problems. Master's thesis, The Chinese University of Hong Kong, Hong Kong (2012)
8. Chudnovsky, M., Seymour, P.D.: Claw-free graphs. IV. Decomposition theorem. J. Comb. Theory, Ser. B **98**(5), 839–938 (2008). doi:10.1016/j.jctb.2007.06.007
9. Chudnovsky, M., Seymour, P.D.: Claw-free graphs. V. Global structure. J. Comb. Theory, Ser. B **98**(6), 1373–1410 (2008). doi:10.1016/j.jctb.2008.03.002
10. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015). doi:10.1007/978-3-319-21275-3
11. Cygan, M., Kowalik, L., Pilipczuk, M.: Open Problems from Workshop on Kernels. Available at http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf (2013)
12. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer (1999). doi:10.1007/978-1-4612-0515-9
13. Drange, P.G., Fomin, F.V., Pilipczuk, M., Villanger, Y.: Exploring the subexponential complexity of completion problems. TOCT **7**(4), 14 (2015). doi:10.1145/2799640
14. Drange, P.G., Pilipczuk, M.: A polynomial kernel for trivially perfect editing. In: Bansal, N., Finocchi, I. (eds.) Algorithms - ESA 2015, 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings, Lecture Notes in Computer Science, vol. 9294, pp. 424–436. Springer (2015). doi:10.1007/978-3-662-48350-3_36
15. Fellows, M., Heggernes, P., Rosamond, F.A., Sloper, C., Telle, J.A.: Finding k disjoint triangles in an arbitrary graph. In: Hromkovic, J., Nagl, M., Westfechtel, B. (eds.) Graph-Theoretic Concepts in Computer Science, 30th International Workshop, WG 2004. Revised Papers, Lecture Notes in Computer Science, vol. 3353, pp. 235–244. Springer, Bad Honnef (2004). doi:10.1007/978-3-540-30559-0_20
16. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin Heidelberg (2006). doi:10.1007/3-540-29953-X
17. Fomin, F.V., Saurabh, S., Villanger, Y.: A polynomial kernel for proper interval vertex deletion. SIAM J. Discrete Math. **27**(4), 1964–1976 (2013). doi:10.1137/12089051X
18. Fomin, F.V., Villanger, Y.: Subexponential parameterized algorithm for minimum fill-in. SIAM J. Comput. **42**(6), 2197–2216 (2013)
19. Ghosh, E., Kolay, S., Kumar, M., Misra, P., Panolan, F., Rai, A., Ramanujan, M.: Faster parameterized algorithms for deletion to split graphs. Algorithmica (2013). doi:10.1007/s00453-013-9837-5
20. Guillemot, S., Havet, F., Paul, C., Perez, A.: On the (non-)existence of polynomial kernels for P_l-free edge modification problems. Algorithmica **65**(4), 900–926 (2013)
21. Hermelin, D., Mnich, M., van Leeuwen, E.J.: Parameterized complexity of induced graph matching on claw-free graphs. Algorithmica **70**(3), 513–560 (2014)
22. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. Syst. Sci. **63**(4), 512–530 (2001). doi:10.1006/jcss.2001.1774
23. Kloks, T., Kratsch, D., Müller, H.: Dominoes. In: Mayr, E.W., Schmidt, G., Tinhofer, G. (eds.) Graph-Theoretic Concepts in Computer Science, 20th International Workshop, WG '94 Herrsching, Germany, June 16-18, 1994. Proceedings, Lecture Notes in Computer Science, vol. 903, pp. 106–120. Springer (1994). doi:10.1007/3-540-59071-4_41
24. Komusiewicz, C., Uhlmann, J.: Cluster editing with locally bounded modifications. Discret. Appl. Math. **160**(15), 2259–2270 (2012)
25. Kratsch, S., Wahlström, M.: Two edge modification problems without polynomial kernels. Discret. Optim. **10**(3), 193–199 (2013). doi:10.1016/j.disopt.2013.02.001
26. Metelsky, Y., Tyshkevich, R.: Line graphs of Helly hypergraphs. SIAM J. Discrete Math. **16**(3), 438–448 (2003). doi:10.1137/S089548019936521X
27. Paul, C., Perez, A., Thomassé, S.: Conflict packing yields linear vertex-kernels for k -fast, k -dense RTI and a related problem. In: Murlak, F., Sankowski, P. (eds.) Proceedings of Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011. Lecture Notes in Computer Science, vol. 6907, pp. 497–507. Springer, Warsaw (2011). doi:10.1007/978-3-642-22993-0_45
28. Yannakakis, M.: Edge-deletion problems. SIAM J. Comput. **10**(2), 297–309 (1981). doi:10.1137/0210021