On the (non-)existence of polynomial kernels for P_l-free edge modification problems

Sylvain Guillemot, Frédéric Havet, Christophe Paul, Anthony Perez

Noname manuscript No.
(will be inserted by the editor)

On the (non-)existence of polynomial kernels for P_l-free edge modification problems

Sylvain Guillemot · Frédéric Havet · Christophe Paul · Anthony Perez

Received: date / Accepted: date

Abstract Given a graph $G = (V, E)$ and a positive integer k, an edge modification problem for a graph property Π consists in deciding whether there exists a set F of pairs of V of size at most k such that the graph $H = (V, E \vartriangle F)$ satisfies the property Π. In the Π *edge-completion problem*, the set F is constrained to be disjoint from E; in the Π *edge-deletion problem*, F is a subset of E; no constraint is imposed on F in the Π *edge-editing problem*. A number of optimization problems can be expressed in terms of graph modification problems which have been extensively studied in the context of parameterized complexity [5,10,16]. When parameterized by the size k of the set F, it has been proved that if Π is an hereditary property characterized by a finite set of forbidden induced subgraphs, then the three Π edge-modification problems are FPT [5]. It was then natural to ask [5] whether these problems also admit a polynomial kernel. Using recent lower bound techniques, Kratsch and Wahlström answered this question negatively [18]. However, the problem remains open on many natural graph classes characterized by forbidden induced subgraphs. Kratsch and Wahlström asked whether the result holds when the forbidden subgraphs are paths or cycles and pointed out that the problem is already open in the case of P_4-free graphs (i.e. cographs). This paper provides positive and negative results in that line of research. We prove that PARAMETERIZED COGRAPH EDGE-MODIFICATION problems have cubic vertex kernels whereas polynomial kernels are unlikely to exist for the P_l-FREE EDGE-DELETION and the C_l-FREE EDGE-DELETION problems for $l \geq 7$ and $l \geq 4$ respectively. Indeed, if they exist, then $NP \subseteq coNP/poly$.

Sylvain Guillemot
Lehrstuhl für Bioinformatik, Friedrich-Schiller Universität Jena

Frédéric Havet
CNRS, Université Nice-Sophia Antipolis, INRIA Sophia-Antipolis

Christophe Paul, Anthony Perez
Université Montpellier II - CNRS, LIRMM E-mail: {paul,perez}@lirmm.fr

Keywords parameterized complexity · kernelization algorithms · cograph · modular decomposition

1 Introduction

An edge modification problem aims at changing the edge set of an input graph $G = (V, E)$ in order to get a certain property Π satisfied (see [20] for a recent study). Edge modification problems cover a broad range of graph optimization problems among which completion problems (*e.g.* MINIMUM FILL-IN, *a.k.a* CHORDAL GRAPH COMPLETION [24, 26]), editing problems (*e.g.* CLUSTER EDITING [25]) and edge deletion problems (*e.g.* MAXIMUM PLANAR SUBGRAPH [13]). In a completion problem, the set F of modified edges is constrained to be disjoint from E; in an edge deletion problem, F has to be a subset of E; and in an editing problem, no restriction applies to F. These problems are fundamental in graph theory and play an important role in computational complexity theory (indeed they represent a large number of the earliest NP-Complete problems [13]). Edge modification problems are also relevant in the context of applications as graphs are often used to model data sets which may contain errors. Adding or deleting an edge thereby corresponds to fixing some false negatives or false positives (see *e.g.* [25] in the context of CLUSTER EDITING). Different variants of edge modification problems have been studied in the literature such as graph sandwich problems [14]. Most of the edge modification problems turn out to be NP-Complete [20] and approximation algorithms exist for some known graph properties (see *e.g.* [17, 27]). But in order to compute an exact solution, fixed parameter algorithms [8, 11, 21] are a good alternative to cope with such hard problems. In the last decades, edge modification problems have been extensively studied in the context of fixed parameterized complexity (see [5, 10, 16]).

A parameterized problem Q is *fixed parameter tractable* (FPT for short) with respect to parameter k whenever it can be solved in time $f(k) \cdot n^{O(1)}$, where $f(k)$ is an arbitrary computable function [8, 21]. In the context of edge modification problems, the size k of the set F of modified edges is a natural parameterization. The generic question is thereby whether a given edge modification problem is FPT for this parameterization. More formally:

PARAMETERIZED Π EDGE–MODIFICATION PROBLEM:
Input: An undirected graph $G = (V, E)$.
Parameter: An integer $k \geqslant 0$.
Question: Is there a subset $F \subseteq V \times V$ with $|F| \leqslant k$ such that the graph $H = (V, E \vartriangle F)$ satisfies Π?

A classical result of parameterized complexity states that a parameterized problem Q is FPT if and only if it admits a *kernelization* [21]. A *kernelization* of a parameterized problem Q is a polynomial-time algorithm $\mathcal{K}$ that given an instance (x, k) computes an equivalent instance $\mathcal{K}(x, k) = (x', k')$ such that the sizes of x' and k' are bounded by a computable function $h()$ depending only on the parameter k. The reduced instance (x', k') is called a *kernel* and we say that Q admits a *polynomial kernel* if the function $h()$ is a polynomial. The equivalence between the existence of an FPT algorithm and the existence of a kernelization only yields kernels of (at least) exponential size. Determining whether an FPT problem has kernel of polynomial (or even linear) size is thus an important challenge. Indeed, the existence of such polynomial-time reduction algorithm (or pre-processing algorithm or *reduction rules*) really speed-up the resolution of the problem, especially if it is interleaved with other techniques [22]. However, recent results proved that it is unlikely that every fixed parameter tractable problem admits a polynomial kernel [1].

Cai [5] proved that if Π is an hereditary graph property characterized by a finite set of forbidden subgraphs, then the PARAMETERIZED Π EDGE-MODIFICATION problems (edge-completion, edge-deletion and edge-editing) are FPT. It was then natural to ask [5] whether these Π edge-modification problems also admit a polynomial kernel. Using recent lower-bound techniques, Kratsch and Wahlström answered negatively this question [18]. However, the problem remains open on many natural graph classes characterized by forbidden induced subgraphs. Kratsch and Wahlström asked whether the result holds when the forbidden subgraphs are paths or cycles and pointed out that the problem is already open in the case of P_4-free graphs (i.e. cographs). In this paper, we prove that PARAMETERIZED COGRAPH EDGE MODIFICATION problems have cubic vertex kernels whereas polynomial kernels are unlikely to exist for the P_l-FREE EDGE-DELETION and C_l-FREE EDGE-DELETION problems for large enough l. The NP-Completeness of the cograph edge-deletion and edge-completion problems have been proved in [9].

Outline of the paper. We begin with some notations and definitions regarding parameterized complexity and modular decomposition. We then establish structural properties of optimal edge-modification sets with respect to modules of the input graph. These properties allow us to design general reduction rules for the PARAMETERIZED COGRAPH EDGE-MODIFICATION problems (Section 3.1). We then establish cubic kernels for these problems using an extra sunflower rule (Section 3.2 and 3.3). Finally, we show it is unlikely that the C_l-FREE EDGE-DELETION and P_l-FREE EDGE-DELETION problems admit polynomial kernels (Section 4).

2 Preliminaries

2.1 Notations

We only consider finite undirected graphs without loops nor multiple edges. Given a graph $G = (V, E)$, we denote by xy the edge of E between the vertices x and y of V. We set $n = |V|$ and $m = |E|$ (subscripts may be used to avoid possible confusion). The neighbourhood of a vertex x is denoted by $N(x)$. Two subsets of vertices X and Y are *adjacent* if there exist $x \in X$ and $y \in Y$

such that x and y are adjacent. If S is a subset of vertices, then $G[S]$ is the subgraph induced by S (i.e. any edge $xy \in E$ between vertices $x, y \in S$ belongs to $E_{G[S]}$). Given a set of pairs of vertices F and a subset $S \subseteq V$, $F[S]$ denotes the pairs of F with both vertices in S. Given two sets S and S', we denote by $S \vartriangle S'$ their symmetric difference. Finally, given any integer l, an induced path (resp. cycle) on l vertices is denoted by P_l (resp. C_l).

2.2 Fixed parameter complexity and kernelization

We let Σ denote a finite alphabet and $\mathbb{N}$ the set of natural numbers. A *(classical) problem* Q is a subset of Σ^*, and a string $x \in \Sigma^*$ is an *input* of Q. A *parameterized problem* Q over Σ is a subset of $\Sigma^* \times \mathbb{N}$. The second component of an instance (x, k) of a parameterized problem is called the *parameter*. Given a parameterized problem Q, one can derive its unparameterized (or classical) version $\tilde{Q}$ by $\tilde{Q} = \{x\#1^k : (x, k) \in Q\}$, where # is a symbol that does not belong to Σ.

A parameterized problem Q is *fixed parameter tractable* (FPT for short) if there is an algorithm which given an instance $(x, k) \in \Sigma^* \times \mathbb{N}$ decides whether $(x, k) \in Q$ in time $f(k) \cdot n^{O(1)}$ where $f(k)$ is an arbitrary computable function (see [8, 11, 21]). A *kernelization* of a parameterized problem Q is a polynomial-time algorithm $\mathcal{K}$ which given an instance $(x, k) \in \Sigma^* \times \mathbb{N}$ outputs an instance $(x', k') \in \Sigma^* \times \mathbb{N}$ such that

1. $(x, k) \in Q$ if and only if $(x', k') \in Q$, and
2. $|x'|, k' \leqslant h(k)$ for some computable function $h : \mathbb{N} \to \mathbb{N}$.

The reduced instance (x', k') is called a *kernel* and we say that Q admits a *polynomial kernel* if the function $h()$ is a polynomial. It is well known that a parameterized problem Q is FPT if and only if it has a kernelization [21]. But this equivalence only yields (at least) exponential size kernels. Recent results proved that it is unlikely that every fixed parameter tractable problem admits a polynomial kernel [1]. These results rely on the notion of *(or-)composition algorithms* for parameterized problems, which together with a polynomial kernel would imply a collapse in the polynomial hierarchy [1]. An *or-composition algorithm* for a parameterized problem Q is an algorithm that receives as input a sequence of instances $(x_1, k) \dots (x_t, k)$ with $(x_i, k) \in \Sigma^* \times \mathbb{N}$ for $1 \leqslant i \leqslant t$, runs in time polynomial in $\sum_{i=1}^{t} |x_i| + k$ and outputs an instance $(y, k') \in \Sigma^* \times \mathbb{N}$ such that:

1. $(y, k') \in Q$ if and only if $(x_i, k) \in Q$ for some $1 \leqslant i \leqslant t$, and
2. k' is polynomial in k.

A parameterized problem admitting an *or-composition algorithm* is said to be *or-compositional*.

Theorem 1 *[1, 12] Let Q be an or-compositional parameterized problem whose unparameterized version $\tilde{Q}$ is NP-complete. The problem Q does not admit a polynomial kernel unless $NP \subseteq coNP/poly$.*

Let P and Q be parameterized problems. A *polynomial-time-and-parameter transformation* from P to Q is a polynomial-time computable function $\mathcal{T}$ which, given an instance $(x,k) \in \Sigma^* \times \mathbb{N}$, outputs an instance $(x',k') \in \Sigma^* \times \mathbb{N}$ such that

1. $(x,k) \in P$ if and only if $(x',k') \in Q$, and
2. $k' \leqslant p(k)$ for some polynomial p.

Theorem 2 ([3]) *Let P and Q be parameterized problems and let $\tilde{P}$ and $\tilde{Q}$ be their unparameterized versions. Suppose that $\tilde{P}$ is NP-complete and $\tilde{Q}$ belongs to NP. If there is a polynomial-time-and-parameter transformation from P to Q and if Q admits a polynomial kernel, then P also admits a polynomial kernel.*

2.3 Modular decomposition and cographs

A *module* in a graph $G = (V,E)$ is a set of vertices $M \subseteq V$ such that for any $x \notin M$ either $M \subseteq N(x)$ or $M \cap N(x) = \emptyset$. Clearly if $M = V$ or $|M| = 1$, then M is a module. We call such a module *trivial*. A graph without any non-trivial module is called *prime*. For two disjoint modules M and M', either all the vertices of M are adjacent to all the vertices of M' or none of the vertices of M is adjacent to any vertex of M'. A partition $\mathcal{P} = \{M_1,\dots,M_p\}$ of the vertex set $V(G)$ whose parts are modules is a *modular partition*. A *quotient graph* $G_{/\mathcal{P}}$ is associated with any modular partition $\mathcal{P}$: its vertices are the parts of $\mathcal{P}$ and there is an edge between M_i and M_j if and only if M_i and M_j are adjacent in G.

A module M is *strong* if for any module M' distinct from M, either $M \cap M' = \emptyset$ or $M \subset M'$ or $M' \subset M$. It is clear from definition that the family of strong modules arranges in an inclusion tree, called the *modular decomposition tree* and denoted $MD(G)$. Each node N of $MD(G)$ thereby represents the set of leaves (vertices of G) for which N is an ancestor. With every node N of $MD(G)$ is associated a quotient graph G_N whose vertices correspond to the children $N_1,\dots,N_p$ of N (see Figure 1 for an example): i.e. $G_N = G[N]_{/\{N_1,\dots,N_p\}}$. We say that a node N of $MD(G)$ is *parallel* if G_N has no edge, *series* if G_N is complete, and *prime* if G_N is prime. A strong property of the family of modules in a graph (used in Lemma 4) is that every module M is either a strong module or there exists a series or a parallel node N such that M is the union of strong modules represented by a subset of the children of N. For a survey on modular decomposition theory, refer to [15].

Definition 1 Let $G_i = (V_i,E_i)$, $1 \leq i \leq k$ be vertex-disjoint graphs. The *series composition* or *join* of the G_i, $1 \leq i \leq k$ is the graph $G_1 \oplus \cdots \oplus G_k = (\bigcup_{i\in I} V_i, \bigcup_{i\in I} E_i \cup \{v_iv_j \mid v_i \in G_i, v_j \in G_j \text{ and } i \neq j\})$. The *parallel composition* (or *disjoint union*) of G_1 and G_2 is the graph $G_1 + \cdots + G_k = (\bigcup_{i=1}^k V_i, \bigcup_{i=1}^k E_i)$.

Fig. 1 A graph G and its modular decomposition tree $MD(G)$. The root of $MD(G)$ is prime and its quotient graph is the 5 vertex graph depicted beside. Every other node is either parallel or series.

Parallel and series nodes in the modular decomposition tree respectively correspond to a parallel and series composition of their children.

Cographs are known as P_4-free graphs, i.e. a cograph does not contain any induced P_4 (see [4,15] for example). However, they were originally defined as follows:

Definition 2 ([4]) A graph is a cograph if it can be constructed from single vertex graphs by a sequence of parallel and series composition.

In particular, this means that the modular decomposition tree of a cograph does not contain any prime node. It follows that cographs are also known as the totally decomposable graphs for the modular decomposition.

3 Polynomial kernels for cograph modification problems

3.1 Modules in optimal solutions

Since cographs correspond to P_4-free graphs, cograph edge-modification problems consist in adding or deleting at most k edges to the input graph in order to make it P_4-free. The use of the modular decomposition tree in our algorithms follows from the following observation.

Observation 1 (Folklore) *Let M be a module of a graph $G = (V, E)$ and $\{a, b, c, d\}$ be four vertices inducing a P_4 of G, then $|M \cap \{a, b, c, d\}| \leqslant 1$ or $\{a, b, c, d\} \subseteq M$.*

This means that given a modular partition $\mathcal{P}$ of a graph G, any induced P_4 of G is either contained in some part of $\mathcal{P}$ or intersects the parts of $\mathcal{P}$ in at most one vertex. This observation allows us to show that a cograph edge-modification problem can be solved independently on modules of the partition $\mathcal{P}$ and on the quotient graph $G_{/\mathcal{P}}$, as stated by the following results.

Observation 2 *Let M be a non-trivial module of a graph $G = (V, E)$. Let F_M be an optimal edge-deletion (resp. edge-completion, edge-edition) set of $G[M]$ and let F_{opt} be an optimal edge-deletion (resp. edge-completion, edge-edition) set of G. Then*

$$F = (F_{opt} \setminus F_{opt}[M]) \cup F_M$$

is an optimal edge-deletion (resp. edge-completion, edge-edition) set of G.

Proof By Observation 1, it follows that $H = (V, E \vartriangle F)$ is P_4-free, thereby F is an edge-deletion set. As being a cograph is an hereditary property, $F_{opt}[M]$ is an edge-deletion set of $G[M]$. Now observe that $|F| = |F_{opt}|$ since otherwise $|F_M| > |F_{opt}[M]|$, which would contradict the optimality of F_M. The same argument holds for edge-completion and edge-edition sets. □

Lemma 3 *Let M be a module of a graph $G = (V, E)$. There exists an optimal edge-deletion (resp. edge-completion, edge-edition) set F such that M is a module of the cograph $H = (V, E \vartriangle F)$.*

Proof Let F_{opt} be an optimal edge-deletion set and denote $H_{opt} = (V, E \vartriangle F_{opt})$. Let x be a vertex of M such that $|\{xy \in F : y \notin M\}|$ is minimum. We argue that the following set of edges is an optimal edge-deletion set:

$$F = F_{opt}[M] \cup F_{opt}[V \setminus M] \cup \{zy : z \in M, y \notin M, xy \in F_{opt}\}$$

First observe that by construction M is a module in the graph $H = (V, E \vartriangle F)$ and that by the choice of x, $|F| \leqslant |F_{opt}|$. Let us prove that H is P_4-free. As $H[M]$ and $H[V \setminus M]$ are respectively isomorphic to $H_{opt}[M]$ and $H_{opt}[V \setminus M]$, they are P_4-free. So if H contains an induced P_4, its vertex set $\{a, b, c, d\}$ intersects M and $V \setminus M$. As M is a module of H it follows by Observation 1 that $|M \cap \{a, b, c, d\}| = 1$ (say $a \in M \cap \{a, b, c, d\}$). It follows by construction of F, that $\{x, b, c, d\}$ also induces a P_4 in H_{opt}, contradicting the assumption that F_{opt} is an edge-deletion set. So we proved that F is an edge-deletion set of G which preserves the module M and is not larger than F_{opt}. The same proof holds for edge-completion and edge-edition sets. □

We want to prove the existence of an optimal solution F (for the edge-deletion, edge-completion and edge-editing problems) that *preserves* every module of G: that is every module of G is a module of $H = (V, E \vartriangle F)$. To that aim Lemma 3 is not yet enough. But if we apply it in a bottom-up manner on the strong modules in the modular decomposition tree, we can prove that such an optimal solution does exist.

Lemma 4 *Let $G = (V, E)$ be an arbitrary graph. There exists an optimal edge-deletion (resp. edge-completion, edge-edition) set F such that every module M of G is a module of the cograph $H = (V, E \vartriangle F)$.*

Proof We prove the statement for edge-deletion sets by induction on the number of non-trival strong modules of a graph. The same proof applies for edge-completion and edge-edition sets. Observe that the result trivially holds if G is a prime graph. Suppose that G has a unique non-trivial strong module M. If M is prime, then the result follows from Lemma 3 as G contains a unique non-trivial module. Assume M is represented by a series or a parallel node. Then the non-trivial modules of G (distinct from M) are exactly the non-trivial subsets of M (see Subsection 2.3). Moreover as $G[M]$ is either a clique or an independent set, it is P_4-free. By Observation 2, there is no need to change the edge set of $G[M]$. The result follows.

Let us now assume that the property holds for every graph with at most t non-trivial strong modules. Let G be a graph with $t+1$ non-trivial strong modules and let M be a non-trivial strong module of G which is minimal for inclusion. By induction hypothesis, the statement holds on $G[M]$ (since it is a prime graph or a clique or an independent set) and on the graph $G_{M \to x}$ where M has been contracted to a single vertex x (since it contains at most t non-trivial strong modules). The conclusion follows from Observation 2. □

As a direct corollary we obtain the existence of an optimal solution that either changes all or none of the edges between two disjoint modules.

Corollary 1 *Let $G = (V, E)$ be an arbitrary graph. There exists an optimal edge-deletion (resp. edge-completion, edge-edition) set F such that for every pair M and M' of disjoint modules, either $(M \times M') \subseteq F$ or $(M \times M') \cap F = \emptyset$.*

3.2 Dismantling the modular decomposition tree

We now present three reduction rules which apply to the three cograph edge-modification problems we consider. The second reduction rule is not required to obtain a polynomial kernel for each of these problems. However, it will ease the analysis of the structure of a reduced graph. The idea behind these rules is to simplify the modular decomposition tree of the input graph. The modular decomposition tree of the reduced graph will have depth at most two.

The three following reduction rules preserve the parameter and only modify the graph.

Rule 1 *Remove the connected components of G which are cographs.*

Rule 2 *If $C = G_1 \oplus G_2$ is a connected component of G, then replace C by $G_1 + G_2$.*

Rule 3 *If M is a non-trivial module of G which is strictly contained in a connected component and is not an independent set of size at most $k+1$, then return the graph $G' + G[M]$ where G' is obtained from G by deleting M and adding an independent set of size $\min\{|M|, k+1\}$ having the same neighbourhood than M.*

Observe that if $G[M]$ is a cograph, adding a disjoint copy to the graph is irrelevant since it will then be removed by Rule 1.

A reduction rule of a parameterized problem Q is said to be *safe* if for any instance (x, k), the rule applied to it returns an equivalent instance (x', k'), (that is $(x, k) \in Q$ if and only if $(x', k') \in Q$).

Lemma 5 *Rules 1, 2 and 3 are safe and can be carried out in linear time.*

Proof The three rules can be computed in linear time using any linear-time modular-decomposition algorithm [15]. The first rule is trivially safe. The second rule is safe by Lemma 4. The safeness of Rule 3 follows from Corollary 1:

there always exists an optimal solution that edits all or none of the edges between any two disjoint modules. Thereby if a module M has size larger than $k+1$, none of the edges (or non-edges) xy with $x \in M$, $y \notin M$ can be changed in such a solution. Shrinking M into an independent set of size $k+1$ and adding a disjoint copy of $G[M]$ (to keep track of the edge modifications inside the module) is thereby safe. □

The analysis of the size of the kernel relies on the following structural property of the modular decomposition tree of an instance reduced under Rule 1, Rule 2 and Rule 3.

Observation 6 *Let G be a graph reduced under Rule 1, Rule 2 and Rule 3. If C is a non-prime connected component of G, then the modules of C are independent sets of size at most $k+1$.*

Proof By Rule 2, none of the connected components of G results from a series composition. By Rule 3, a module strictly contained in C has size at most $k+1$ and is an independent set. □

Observe that Rule 3 increases the number of vertices of the instance. Nevertheless, we will be able to bound the number of vertices of a reduced instance.

It remains to show that computing a reduced graph requires polynomial time. Let us mention that it is safe to apply Rule 2 and Rule 3 only on strong modules (in Rule 2, G_1 can be chosen as a strong module). This will optimize the number of rules applications.

Lemma 7 *Given a graph $G = (V, E)$, computing a graph reduced under Rule 1, Rule 2 and Rule 3 requires polynomial time.*

Proof Let us say that a module M of G is *reduced* if it is an independent set of size at most $k+1$ or the disjoint union of some connected components of G (observe that connected components of G are also modules of G). By Observation 6, if G is reduced under Rule 1, Rule 2 and Rule 3, then every module of G is reduced. Notice that if every strong module of G is reduced, then every module of G is reduced. So to prove the statement, we count the number of strong modules (*i.e.* nodes of the modular decomposition tree $MD(G)$) which are not reduced.

Let us also remark that if a connected component C is a cograph with at least two vertices, then a series of applications of Rule 2 eventually transforms C in a set of isolated vertices. This means that we can assume that the applications of Rule 1 is postponed to the end of the reduction process. This will ease the argument below.

When Rule 3 is applied on a strong module, then by definition the number of non-reduced strong modules decreases by one. When Rule 2 is applied (i.e. $G = G_1 \oplus G_2$ and G_1 is induced by a strong module), unless G_1 is an independent set of size at most $k+1$, then the number of non-reduced strong modules also decreases by one. But observe that if G_1 is an independent set of size at most $k+1$, then its vertices will be removed by Rule 1 as they will become isolated vertices. As the number of strong modules of a graph is bounded by the number of vertices, this proves that a series of at most n applications of Rule 2 and Rule 3 is enough to compute a reduced graph. $\square$

3.3 Cograph edge-deletion (and edge-completion)

In addition to the previous reduction rules, we need the classical *sunflower* rule to obtain a polynomial kernel for the PARAMETERIZED COGRAPH EDGE-DELETION problem.

Rule 4 *If e is an edge of G that belongs to a set $\mathcal{P}$ of at least $k+1$ P_4's such that e is the only common edge of any two distinct P_4's of $\mathcal{P}$, then remove e and decrease k by one.* (See Figure 2.)

Fig. 2 The two distinct cases of a sunflower of 5 edge-disjoint P_4. Edges e and e' have to be removed when $k \leq 4$.

Observation 8 *Rule 4 is safe and can be carried out in polynomial time.*

Proof It is clear that the edge e has to be deleted as otherwise at least $k+1$ edge deletions would be required to break all the P_4's of the set $\mathcal{P}$. Such an edge, if it exists, can be found in polynomial time if one computes the set of all P_4's of the input graph (which can be done in $O(n^4)$ time). $\square$

To analyse the size of a reduced graph $G = (V, E)$, we study the structure of the cograph $H = (V, E \bigtriangleup F)$ resulting from the removal of an optimal (of size at most k) edge-deletion set F. The modular decomposition tree or *cotree* is the appropriate tool for this analysis.

Theorem 3 *The* PARAMETERIZED COGRAPH EDGE-DELETION *problem admits a cubic vertex kernel.*

Proof Let $G = (V, E)$ be a graph reduced under Rule 1, Rule 2, Rule 3 and Rule 4 that can be turned into a cograph by deleting at most k edges. Let F be an optimal edge-deletion set and denote by $H = (V, E \bigtriangleup F)$ the cograph resulting from the deletion of F and by T its cotree. We will count the number of leaves of T (or equivalently of vertices of G and H).

Observe that since a set of k edges covers at most $2k$ vertices, T contains at most $2k$ *affected* leaves (i.e. leaves corresponding to a vertex incident to an edge of F). We say that an internal node of the cotree T is *affected* if it is the least common ancestor of two affected leaves. Notice that there are at most $2k$ affected nodes.

We first argue that the root of T is a parallel node and is affected. Assume that the root of T is a series node: since no edges are added to G, this would imply that G is not reduced under Rule 2, a contradiction. Moreover, since G is reduced under Rule 1, none of its connected components is a cograph. It follows that every connected component of G contains a vertex incident to a removed edge, and thus that every subtree attached to the root contains an affected leaf as a descendant. Hence the root of T is an affected node.

Claim 9 *Let p be an affected leaf or an affected node different from the root, and q be the least affected ancestor of p. The path between p and q has length at most $2k+3$.*

Proof. Observe first that the result trivially holds if q is the root of T and p one of its children. In all other cases, let M be the set of leaves descendant of p in T. We claim that M contains a leaf x which is incident to a removed edge xy, with $y \notin M$. If p is an affected leaf, then this is true by definition. Otherwise, if p is an affected node different from the root, assume by contradiction that all the removed edges in M are of the form uv with $u, v \in M$. In particular, this implies that M is a module of G strictly contained in a connected component. By Observation 6, it follows that M is an independent set and hence contains no edges, a contradiction. Let t be the least common ancestor of x and y. The node t is a parallel node which is an ancestor of p and q (observe that we may have $t = q$). Assume by contradiction that the path between x and t in T contains a sequence of $2k+3$ consecutive non-affected nodes. The type of these nodes is alternatively series and parallel. So we can find a sequence $s_1, p_1 \dots s_{k+1}, p_{k+1}$ of consecutive non-affected nodes with s_i (resp. p_i) being the father of p_i (resp. s_{i+1}) and with s_i's being series nodes and the p_i's being parallel node. Now each of the s_i's (resp. p_i) has a non-affected leaf a_i (resp. b_i) which is not a descendant of p_i (resp. s_{i+1}). Observe that for every $i \in [1, k+1]$ the vertex set $\{b_i, a_i, x, y\}$ induces a P_4 in G. Thereby we found a set of $k+1$ P_4's in G pairwise intersecting on the edge xy. It follows that G is not reduced by the Rule 4, a contradiction. Since all nodes between p and q are non-affected, it follows that the path between p and q contains at most $2k+3$ nodes. $\diamond$

Since there are at most $2k$ affected nodes and $2k$ affected leaves, T contains at most $(4k-1)(2k+3)+2k$ internal nodes. As G is reduced, Observation 6 implies that each of these $O(k^2)$ nodes is attached to a set of at most $k+1$ leaves or a parallel node with $k+1$ children. It follows that T contains at most $2k+(k+1)[(4k-1)(2k+3)+2k] \leqslant 8k^3+20k^2+11k$ leaves, which correspond to the number of vertices of G.

We now conclude with the time complexity needed to compute the kernel. Since the application of Rule 4 decreases the value of the parameter (which is not changed by the other rules), Rule 4 is applied at most $k \leqslant n^2$ times. It then follows from Lemma 7 that a reduced instance can be computed in polynomial time. □

The following corollary simply follows from the observation that the family of cographs is closed under complementation (since the complement of a P_4 is a P_4).

Corollary 2 *The* Parameterized cograph edge-completion *problem admits a cubic vertex kernel.*

3.4 Cograph edge-editing

The lines of the proof for the cubic kernel of the edge-editing problem are essentially the same as for the edge-deletion problem. But since edges can be added and deleted, the reduction Rule 4 has to be extended to take into account edges whose addition breaks an arbitrary large set of P_4's.

Rule 5 *If $\{x, y\}$ is a pair of vertices of G that belongs to a set $\mathcal{S}$ of $t \geqslant k+1$ quadruples $P_i = \{x, y, a_i, b_i\}$ such that for every $1 \leqslant i \leqslant t$, P_i induces a P_4 and for any $1 \leqslant i < j \leqslant t$, $P_i \cap P_j = \{x, y\}$, then change E into $E \triangle \{xy\}$ and decrease k by one.*

As for reduction Rule 4, it is clear that reduction Rule 5 is safe and can be applied in polynomial time. The kernelization algorithm of cograph edge-editing consists of an exhaustive application of Rules 1, 2, 3 and 5.

Theorem 4 *The* Parameterized cograph edge-editing *problem admits a cubic vertex kernel.*

Proof Let $G = (V, E)$ be a graph reduced under Rule 1, Rule 2, Rule 3 and Rule 5 that can be turned into a cograph by editing at most k edges. Let H be the cograph obtained by an optimal edge-edition. The cotree of H is denoted by T. Unlike in the edge-deletion problem, the root of T is not necessary a parallel node. However it is still true that the root of T is affected. Indeed, assume first that the root of T is a series node. Then it is affected since otherwise G would not be reduced under Rule 2. Now, assume that the root is a non affected parallel node. This means that at most one of its children contains an affected leaf as descendant, and hence that G is not reduced under Rule 1, a contradiction.

In the following we assume w.l.o.g. that the root of T is a parallel node. We prove that Claim 9 still holds in this case. Let p be an affected leaf or an affected node different from the root, and q be the least affected ancestor of p. Observe that the result is trivially true if q is the root of T and p one of its children. In all other cases, let M be the set of leaves descendant of p in T. As in the proof of Theorem 3, there must exist an edited edge xy with

$x \in M, y \notin M$ (otherwise M would be a module of G, i.e. an independent set by Observation 6 and would thus not be edited by Observation 2).

Now the proof follows the arguments of the proof of Theorem 3, if one can find in T a path of $2k + 3$ consecutive non-affected nodes between p and q, then G is not reduced under Rule 5. Proving that T contains $O(k^2)$ nodes and thereby $O(k^3)$ leaves.

The fact that a reduced instance can be computed in polynomial time follows from Lemma 7 and the observation that Rule 5 decreases the value of the parameter and requires polynomial time. □

For the deletion (resp. editing) problem there exists a graph reduced under Rule 1, Rule 2, Rule 3 and Rule 4 (resp. Rule 5) that achieves the cubic bound (see Figure 3).

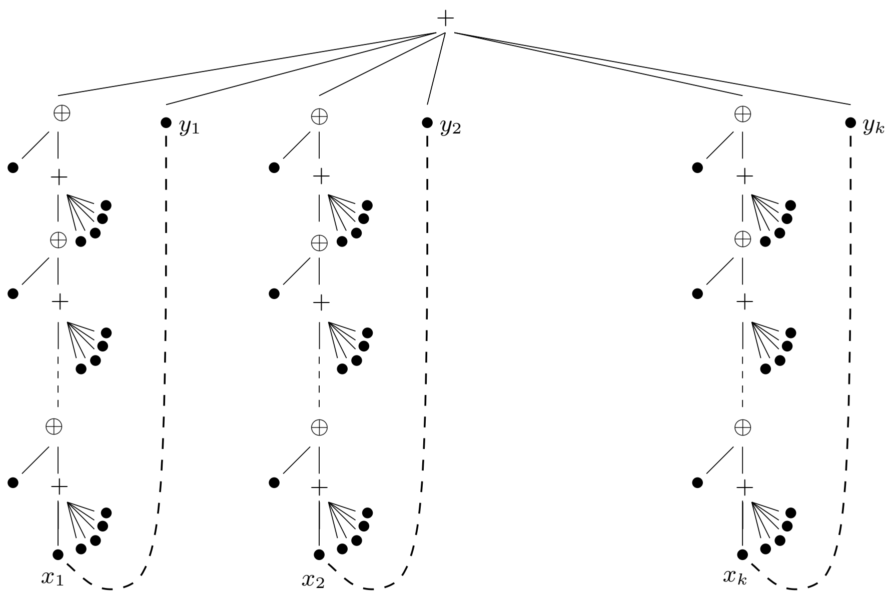

Fig. 3 A reduced graph G with $k(k+1)^2+k$ vertices for which k edge deletions, namely the x_iy_i's for $i \in [1, k]$, are required to obtain a cograph H. The cotree T of H is represented. Each parallel node of T which is not the root has $k + 2$ children, $k + 1$ of which are leaves. The root of T has $2k$ children.

t

4 Kernel lower bounds for P_l-FREE EDGE-DELETION problems

In [18], Kratsch and Wahlström show that the NOT-1-IN-3-SAT problem has no polynomial kernel under the complexity-theoretic assumption $NP \nsubseteq coNP/poly$.

We observe that their argument still applies to a graph restriction of Not-1-in-3-sat where the constraints arise from the triangles of an input graph.

4.1 A graphic version of the Not-1-in-3-sat problem

For a graph $G = (V, E)$, an *edge-bicoloring* is a function $B : E \to \{0, 1\}$. A *partial edge-bicoloring* of G is an edge-bicoloring of a subset of edges of E. An edge colored 1 (resp. 0) is called a 1-*edge* (resp. 0-*edge*). We say that the edge-bicoloring B' *extends* a partial edge-bicoloring B if $B'(e) = B(e)$ for every edge $e \in E$ colored by B. The *weight* of an edge-bicoloring B is the number $\omega(B)$ of 1-edges. An edge-bicoloring is *valid* if every triangle of G contains either zero, two or three 1-edges. We consider the following problem:

Not-1-in-3-edge-triangle:
Input: An undirected graph $G = (V, E)$ and a partial edge-bicoloring $B : E' \to \{0, 1\}$ with $E' \subseteq E$.
Parameter: An integer $k \in \mathbb{N}$.
Question: Can we extend B to a *valid* edge-bicoloring B' of weight at most k?

Proposition 1 Not-1-in-3-edge-triangle *is NP-complete and or-compositional.*

Proof The NP-hardness follows from a reduction from Vertex Cover. Let (G, k) be an instance of Vertex Cover [13], where $G = (V, E)$. We create an instance (G', B, k') of Not-1-in-3-edge-triangle as follows. The graph G' is obtained from G by adding a dominating vertex q, the partial edge-bicoloring B is such that $B(e) = 1$ for every $e \in E$, and we let $k' = |E| + k$. As the triangles of G are monochromatic, the constraints to obtain a valid extension of B are carried by the triangles of the form quv with $uv \in E$. It is easy to observe that (G', B, k') has a valid edge-bicoloring extension of weight k' if and only if G has a vertex cover of size k. As Not-1-in-3-edge-triangle clearly belongs to NP, the NP-completeness follows.

We now show that Not-1-in-3-edge-triangle is or-compositional. The proof closely follows the proof of [18] for Not-1-in-3-sat. We first need the following result:

Claim 10 *Given an instance (G, B, k) of* Not-1-in-3-edge-triangle, *and two positive integers p and k' such that $k' \geq k + p$, we can compute in polynomial time an equivalent instance (G', B', k') of* Not-1-in-3-edge-triangle *such that $\omega(B') = \omega(B) + p$.*

Proof. To build G', we first add to G a set F of p new isolated edges $e_1 \dots e_p$ such that $B'(e_i) = 1$ for all $i \in [p]$. Then we add to the resulting graph $k' - (k + p)$ gadgets one after another as follows: let $e = uv$ be an arbitrary 1-edge of the graph constructed so far; add the triangles uvx, vxy where x and y are new vertices and set $B'(vy) = B'(xy) = 0$. (The successive edges e

are not necessarily distinct.) Observe that in any valid edge-bicoloring of G' extending B', the edge vx is a 0-edge while the edge ux is a 1-edge. It follows that (G,B,k) is a positive instance if and only if (G',B',k') is a positive instance as the set F increases the weight by p and the added triangles by $k'-(k+p)$. $\diamond$

Consider a sequence $(G_1,B_1,k),\ldots,(G_t,B_t,k)$ of instances of Not-1-in-3-edge-triangle. Free to remove the instances such that $w(B_j)>k$, which are trivially false, we may assume that $w(B_j)\leq k$ for $1\leq j\leq t$. Furthermore, by Claim 10, we can assume w.l.o.g. that $w(B_j)=k$, for $1\leqslant j\leqslant t$.

We can also assume that $t\leqslant 2^k$ since otherwise an easy exact branching algorithm solves the problem. Finally, for the sake of the construction, we assume $t=2^l$ (free to duplicate some instances (G_i,B_i,k) if necessary).

We denote by $E_1(j)$ the set of 1-edges of (G_j,B_j,k).

Intuitively, the graph G of the composed instance (G,B,k') is built on the disjoint union of the G_j's, $1\leqslant j\leqslant t$. Then, as a selection gadget, we add a "tree-like graph" T connecting a"root edge" e_r to edges e_j for $j=1,...,t$. Finally, for every $1\leqslant j\leqslant t$, the 1-edges of the graph G_j are connected via a propagation gadget to the edge e_j in T. The root edge is the unique 1-edge of G. The copies of the G_j's inherit the 0-edges of the G_j's, the other edges are uncolored. The idea is that the selection gadget guarantees that at least one of the e_j's edge gets colored 1. Then the propagation gadgets attached to that edge e_j transmit color 1 to the copies of every 1-edge of G_j.

Formally, we do the following: (i) we start with a complete binary tree T_0 with root r and t leaves $s_1,\ldots,s_t$; (ii) to each node u of T_0, we associate an edge e_u in T as follows: if u is associated to the edge xy and if u has two children v,v', we create a new vertex z and we let $e_v=xz, e_{v'}=yz$. For convenience, we write e_j instead of e_{s_j}. Now, for every $1\leqslant j\leqslant t$, the propagation gadget S_j consists of vertex-disjoint graphs $S_{j,e}$ for every edge e of $E_1(j)$. If $e=uv$ and $e_j=xy$, then $S_{j,e}$ consists of four triangles uva, vab, abx, bxy, with edges ua, vb, ax, by colored 0 by B (the other edges remain uncolored). Again the unique 1-edge of B is the root edge of T, in particular the edges of $E_1(j)$ are uncolored by B. However, the 0-edge sets of the G_j's are inherited by B. See Figure 4.

Observe first that every valid edge-bicoloring extending B has to assign color 1 to at least one edge e_j, for $1\leqslant j\leqslant t$, and to the l edges e_v for v vertices of the (r,s_j)-path in T_0. Then the edges of $E_1(j)$ and the $3k$ non 0-edges of S_j are also assigned color 1. Reciprocally, for every $1\leqslant j\leqslant t$, one can extend B in such a way that the only 1-edges are the ones corresponding to the (r,s_j)-path in T_0, the non 0-edges of S_j and some in G_j. Hence, if we choose $k'=k+3k+l$, then (G,B,k') is a positive instance if and only if there exists $1\leqslant j\leqslant t$ such that (G_j,B_j,k) is a positive instance. □

The following corollary follows from Theorem 1:

Corollary 3 *The* Not-1-in-3-edge-triangle *problem does not admit a polynomial kernel unless $NP\subseteq coNP/poly$.*

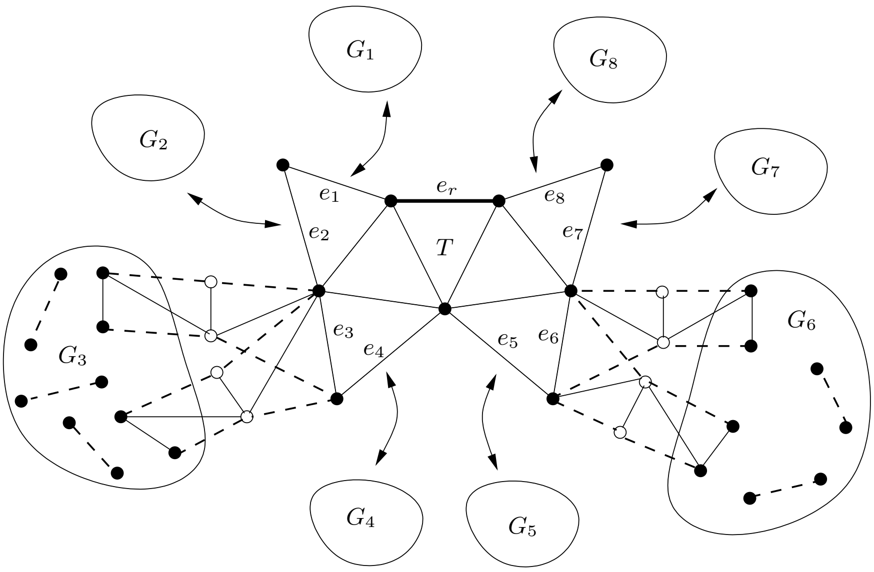

Fig. 4 The instance (G, B, k') built from a sequence $(G_1, B_1, k), \ldots, (G_t, B_t, k)$ with $t = 2^3$. The unique 1-edge is r. Every "leaf edge" e_j of T is linked to the copies of the 1-edges of (G_j, B_j, k) via the propagation gadget. The 0-edges are depicted as dotted lines: they either belong to a propagation gadget or correspond to a 0-edge of some (G_j, B_j, k).

The problem Tripartite-Not-1-in-3-edge-triangle is the restriction of Not-1-in-3-edge-triangle where the input graph $G = (V, E)$ is *tripartite*, i.e. $V = V_1 \cup V_2 \cup V_3$ where V_i is an independent set, $i = 1, 2, 3$. The hardness results obtained for Not-1-in-3-edge-triangle carry over to this restriction:

Lemma 11 *The* Tripartite-Not-1-in-3-edge-triangle *problem does not admit a polynomial kernel unless $NP \subseteq coNP/poly$.*

Proof The proof uses Theorem 2, that is we provide a polynomial-time-and-parameter transformation from Not-1-in-3-edge-triangle to Tripartite-Not-1-in-3-edge-triangle. By Proposition 1, Not-1-in-3-edge-triangle is NP-complete. Observe that Tripartite-Not-1-in-3-edge-triangle clearly belongs to NP.

Let (G, B, k) be an instance of Not-1-in-3-edge-triangle. We build an instance $(G', B', 6k)$ of Tripartite-Not-1-in-3-edge-triangle in the following way. Suppose that $G = (V, E)$, then G' has vertex set $V' = \{v_1, v_2, v_3 : v \in V\}$, and has edge set $E' = \{u_1v_2, u_1v_3, u_2v_3 : u = v \text{ or } uv \in E\}$. The partial edge-bicoloring B' is defined as follows: $B'(u_iu_j) = 0$ for $1 \leqslant i < j \leqslant 3$; if the edge uv of G is colored, then $B'(u_iv_j) = B(uv)$ for $1 \leqslant i, j \leqslant 3$, $i \neq j$; the other edges of G' are uncolored.

Observe that every valid edge-bicoloring extending B' assigns the same color to the six edges of G' associated with an edge uv of G: indeed, given u_iv_k, u_jv_l $1 \leqslant i, j, k, l \leqslant 3$, if $i = j$ this holds since $B'(v_kv_l) = 0$, if $k = l$

this holds since $B'(u_iu_j) = 0$, and otherwise this follows by transitivity. It is then easy to see that solutions of (G, B, k) and solutions of $(G', B', 6k)$ are in one-to-one correspondence. □

4.2 Negative results for Γ-free edge deletion problems

In this subsection, we prove the following twin theorems.

Theorem 5 *For all $l \geqslant 7$, C_l-FREE EDGE-DELETION has no polynomial kernel unless $NP \subseteq coNP/poly$.*

Theorem 6 *For all $l \geqslant 7$, P_l-FREE EDGE-DELETION has no polynomial kernel unless $NP \subseteq coNP/poly$.*

To prove these theorems, we provide polynomial-time-and-parameter transformations from TRIPARTITE-NOT-1-IN-3-EDGE-TRIANGLE to VERTEX ANNOTATED C_l-FREE EDGE-DELETION and to VERTEX ANNOTATED P_l-FREE EDGE-DELETION. For a graph Γ, the VERTEX ANNOTATED Γ-FREE EDGE-DELETION problem is defined as follows.

VERTEX ANNOTATED Γ-FREE EDGE-DELETION
Input: An undirected graph $G = (V, E)$ and a subset S of vertices.
Parameter: An integer $k \in \mathbb{N}$.
Question: Is there a subset $F \subseteq E \cap (S \times S)$ of size at most k such that $H = (V, E \setminus F)$ is Γ-free?

The edges of $E \cap (S \times S)$ are said to be *allowed edges*.

Observe that the VERTEX ANNOTATED Γ-FREE EDGE-DELETION problem reduces to the (unannotated) Γ-FREE EDGE-DELETION problem whenever the class of Γ-free graphs is closed under true (resp. false) twin addition: it suffices to add for every vertex $v \in V \setminus S$ a set of $k+1$ true (resp. false) twins. A *true twin* of v is a vertex u adjacent to v with the same closed neighborhood ($N(v) \cup \{v\} = N(u) \cup \{u\}$). A *false twin* of v is a vertex u non-adjacent to v with the same neighborhood ($N(v) = N(u)$). Clearly adding a false or true twin preserves the parameter.

Since the family of C_l-free graphs is closed under false twin addition (for $l \geqslant 5$), VERTEX ANNOTATED C_l-FREE EDGE-DELETION reduces to C_l-FREE EDGE-DELETION, Theorem 5 is a direct consequence of the following statement.

Lemma 12 *For all $l \geqslant 7$,* VERTEX ANNOTATED C_l-FREE EDGE-DELETION *has no polynomial kernel unless $NP \subseteq coNP/poly$.*

Proof We describe a polynomial-time-and-parameter transformation from TRIPARTITE-NOT-1-IN-3-EDGE-TRIANGLE to VERTEX ANNOTATED C_l-FREE EDGE-DELETION. The result then follows Theorem 2 and Lemma 11.

Observe that we can restrict TRIPARTITE-NOT-1-IN-3-EDGE-TRIANGLE to instances (G, B, k) without 0-edges (*i.e.* $B(e) = 1$ whenever it is defined). The reason is that any uncolored edge $e = uw$ of G can be forced to be assigned color 0 in every valid edge-bicoloring extending B by adding to G a set of $k+1$ new vertices $v_1, \dots, v_k$ such that uv_iw, $1 \leqslant i \leqslant k$, is an uncolored triangle. Clearly if e is a 1-edge of an edge-bicoloring B' extending B, B' needs at least $k+1$ 1-edges to be valid: e plus one edge per triangle. The same argument was used in [18] for the Not-1-in-3-sat problem.

Let (G, B, k) be an instance of the Tripartite-Not-1-in-3-edge-triangle problem, where V_1, V_2, V_3 are disjoint independent sets of $G = (V, E)$. The construction of the instance (H, S, k') of Annotated C_l-free edge-deletion works as follows. First the sets V_1, V_2 and V_3 are turned into cliques and the 1-edges of G are removed. Then, for each pair $t = (uw, v)$ such that $\{u, v, w\}$ induces a triangle in G, we create a path P_t of length $l-3$ with endpoints a_t and b_t and join a_t to $V - \{v, w\}$ and b_t to $V - \{u, v\}$. Notice that each triangle of G generates three such paths in H. Finally, every two vertices x and y belonging to different P_t are made adjacent. We denote by $H = (V_H, E_H)$ the resulting graph. See Figure 5. In other words, H is obtained from the disjoint union of G and the join of the P_t, by turning V_1, V_2, V_3 into cliques and adding edges from a_t to $V - \{v, w\}$ and b_t to $V - \{u, v\}$ for each $t = (uw, v)$ such that $\{u, v, w\}$ induces a triangle in G.

To complete the description of (H, S, k') we set $S = V$ and the parameter $k' = k - k_1$ where k_1 is the number of 1-edges of (G, B, k).

Fig. 5 The graph $H = (V_H, E_H)$ built from an instance (G, B, k) of the Tripartite-Not-1-in-3-edge-triangle problem for $l = 12$. The white and the square vertices form the set U of new vertices. The independent sets V_1, V_2 and V_3 of G are turned into cliques. The thick dotted edges are the removed 1-edges of (G, B, k). The non 1-edges of (G, B, k) are preserved in H.

Claim 13 *A subset of vertices $C \subseteq V_H$ induces a cycle of length l if and only if G contains a triangle uvw, with $e = uw$ a 1-edge and uv, vw uncolored edges, such that $C = P_t \cup \{u, v, w\}$ with $t = (e, v)$.*

Proof. By construction, if G contains a triangle uvw with a unique 1-edge $e = uw$, then $C = P_t \cup \{u, v, w\}$ (with $t = (e, v)$) induces a cycle of length l in H (keep in mind that the 1-edges of G are removed from H). Let C be an induced C_l in H. Observe that as V_1, V_2 and V_3 are turned into cliques, $|C \cap V| \leqslant 6$. Thereby C intersects the vertex set U. We claim that $C \cap U$ is included in a single path P_t. Observe first that $C \cap U$ cannot intersect three paths $P_t, P_{t'}, P_{t''}$, since we would find three vertices inducing a C_3, impossible. Suppose now that $C \cap U$ intersects exactly two distinct paths P_t, $P_{t'}$ ($t \neq t'$). If $C \cap U$ contained two vertices in P_t and two vertices in $P_{t'}$, these four vertices would induce a C_4, impossible. If $C \cap U$ contained three vertices in P_t and one vertex in $P_{t'}$, this vertex would have degree at least 3 in C, impossible. Hence, we can assume that $|C \cap P_t| \leq 2$ and $|C \cap P_{t'}| = 1$. The elements of $C \cap P_t$ must be endpoints of P_t, as together with the vertex of $C \cap P_{t'}$ they form either a P_2 or a P_3. But an element of $C \cap P_t$ is nonadjacent to at least $l-4 \geq 3$ vertices of $C \cap V$, and thus cannot be a_t nor b_t by definition of the adjacencies of these vertices. This is a contradiction.

Hence there exists a path P_t, with $t = (e, v)$ and $e = uw$, containing the vertices of $C \cap U$. We then have the following alternative: either (i) $C \cap P_t \subseteq \{a_t, b_t\}$, or (ii) P_t is included in C. In case (i), we have that a_t or b_t is nonadjacent to at least $l-4 \geq 3$ vertices of $C \cap V$, contradicting the definition of the adjacencies. Thus, we are in case (ii), and C consists of the vertices of P_t, together with three extra vertices $x, y, z \in V$, with x adjacent to a_t only, z adjacent to b_t only, and y nonadjacent to a_t and b_t. We have $y = v$ as v is the only vertex of V nonadjacent to a_t, b_t. We then have $x = u$ as x is nonadjacent to b_t, and $z = w$ as z is nonadjacent to a_t. Now the existence of P_t witnesses the existence of the triangle uvw in G. As $uv, wv \in E_H$ and $uw \notin E_H$, uw is the only 1-edge of the triangle uvw. $\diamond$

We now argue for the correctness of the transformation. Suppose that there exists a set F of allowed edges of size at most k' such that $H' = (V_H, E_H \setminus F)$ is C_l-free. Define the edge-bicoloring B' of E as follows: for any edge $e \in E_H \cap E$ $B'(e) = 1$ if $e \in F$, $B'(e) = 0$ otherwise. Moreover, since any edge $e \in E \setminus E_H$ is a 1-edge, we set $B'(e) = 1$ for such edges. As by assumption B does not assign color 0 to any edge, B' extends B and has weight at most $|F| + k_1 \leqslant k' + k_1 = k$. Besides, B' is a valid edge-bicoloring of G. Let $t = (e, v)$ with $e = uw$ be a pair such that $\{u, v, w\}$ induces a triangle in G. If we had $B(uw) = 1$, $B'(uv) = B'(vw) = 0$, we would obtain that $P_t \cup \{u, v, w\}$ induces a C_l in H', impossible. Conversely, suppose that B' is valid edge-bicoloring of weight at most k of G which extends B. Let $F \subseteq E$ be the set of edges such that $B'(e) = 1$ but are uncolored by B. By construction F is a set of allowed edges of H of size at most $k - k_1$. Since B' is a valid edge-bicoloring of G, Claim 13 implies that $H' = (V_H, E_H \setminus F)$ is C_l-free. $\square$

The proof of Theorem 6 is very similar to the one of Theorem 5. Indeed, since the family of P_l-free graphs is closed under true twin addition for $l \geqslant 3$, it follows the following analog of Lemma 12.

Lemma 14 *For all $l \geqslant 7$,* Vertex annotated P_l-free edge-deletion *has no polynomial kernel unless $NP \subseteq coNP/poly$.*

Proof Let (G, B, k) be an instance of the Tripartite-Not-1-in-3-edge-triangle problem without 0-edges and such that V_1, V_2, V_3 are disjoint independent sets of $G = (V, E)$. We modify the construction given in the proof of Lemma 12 to obtain an instance (H, S, k') of Vertex annotated P_l-free edge-deletion problem. The vertex set V_H of H consists of the union of V and a set U of new vertices. The sets V_1, V_2 and V_3 are again turned into cliques and the 1-edges of E are not duplicated in E_H. But for each pair $t = (e, v)$, with $e = uw \in E$ and $v \in V$ such that $\{u, v, w\}$ is a triangle of G, the associated gadget Q_t is no longer a path. Instead, Q_t consists of two paths Q_t^u and Q_t^w, whose lengths are at least 2 and sum to $l-3$ (this is possible since $l \geq 7$). As before for every $t \neq t'$ we add all the edges between vertices of Q_t and $Q_{t'}$. Let a_t be an endpoint of Q_t^u, and let b_t be an endpoint of Q_t^w, then a_t is made adjacent to $V - \{v, w\}$ and b_t is made adjacent to $V - \{u, v\}$. To complete the description of (H, S, k') we set $S = V$ and $k' = k - k_1$ where k_1 is the number of 1-edges of (G, B, k).

The correctness proof of the construction follows the same lines as the proof of Lemma 12. It now relies on the following claim that characterizes the possible induced paths of length l.

Claim 15 *A subset of vertices $Q \subseteq V_H$ induces a path of length l if and only if G contains a triangle uvw, with $e = uw$ a 1-edge and uv, vw uncolored edges, such that $Q = Q_t \cup \{u, v, w\}$ with $t = (e, v)$.*

Proof. By construction, if G contains a triangle uvw with a unique 1-edge $e = uw$, then $Q = Q_t \cup \{u, v, w\}$ (with $t = (e, v)$) induces a path of length l in H (keep in mind that the 1-edges of G are removed from H). Let Q be an induced P_l in H. As in the proof of Claim 13, observe that $|Q \cap V| \leqslant 6$ and thereby Q intersects the vertex set U and that there exists a unique pair $t = (e, v)$ with $e = uw$ such that Q_t contains $Q \cap U$. We then have either (i) $P \cap Q_t \subseteq \{a_t, b_t\}$, or (ii) $|P \cap Q_t^u| \geq 2$, or (iii) $|P \cap Q_t^w| \geq 2$. In case (i) we have that a_t or b_t are nonadjacent to at least $l - 4 \geq 3$ vertices of V, contradicting the definition of the adjacencies. Suppose now that we are in case (ii). Since $|P \cap Q_t^u| \geq 2$, a_t is adjacent to at most one vertex of V in P. As a_t is nonadjacent to only two vertices of V, we obtain that $|P \cap V| \leq 3$. We reach the same conclusion in case (iii) by considering b_t instead of a_t. It follows that $|P \cap Q_t| \geq l - 3$, and this must be an equality. Hence P consists of the vertices of Q_t^u, followed by three vertices $x, y, z \in V$, followed by the vertices of Q_t^w. As in the proof of Claim 13, we obtain that $x = u, y = v, z = w$. Now, the existence of Q_t witnesses the existence of the triangle uvw in G. Moreover the edge uw cannot exist in H, meaning that uw is a 1-edge of (G, B, k). $\square$

4.3 Improved results for C_l-free edge deletion problems

In the previous section, we obtained kernel bounds for the C_l-free edge deletion problems for every $l \geq 7$. We extend these bounds to $4 \leq l < 7$, by using a different proof technique. We rely on the notion of *cross-composition* introduced by [2].

We need the following definitions from [2]. An equivalence relation R on Σ^* is a *polynomial equivalence relation* if and only if the following two conditions hold:

1. The relation R is decidable in polynomial time.
2. For any finite set $S \subseteq \Sigma^*$, the equivalence relation R partitions the elements of S in at most $(\max_{x \in S} |x|)^{O(1)}$ classes.

Let P be a classical problem, and let Q be a parameterized problem. A *cross-composition* from P to Q consists of a polynomial equivalence relation

R, and of an algorithm A that receives as input a sequence of strings $x_1, ..., x_t$ belonging to the same equivalence class of R, runs in time polynomial in $\sum_{i=1}^{t} |x_i|$, and outputs an instance $(y, k') \in \Sigma^* \times \mathbb{N}$ such that:

1. $(y, k') \in Q$ if and only if $x_i \in P$ for some $1 \leq i \leq t$, and
2. k' is polynomial in $\max_{i=1}^{t} |x_i| + \log t$.

Theorem 7 ([2]) *Let P be a classical problem and let Q be a parameterized problem. Suppose that P is NP-hard under Karp reductions and that there is a cross-composition of P into Q. Then Q does not admit a polynomial kernel unless $NP \subseteq coNP/poly$.*

We define Annotated C_l-free edge deletion as follows: given a graph $G = (V, E)$, a set $F \subseteq E$ of allowed edges, can we remove at most k allowed edges in G to obtain a C_l-free graph? Observe that Vertex annotated C_l-free edge deletion is a particular case of Annotated C_l-free edge deletion. Hence, for $l \geq 7$, both problems have no polynomial kernel unless $NP \subseteq coNP/poly$.

We first prove that this annotated version is unlikely to have a polynomial kernel for any $l \geq 4$, and we transfer the result to the unannotated version using a polynomial-time-and-parameter transformation.

Lemma 16 *The problem* Vertex Cover *is NP-hard on graphs of girth at least* 9.

Proof We give a reduction from Vertex Cover. The reduction consists, starting with a graph $G = (V, E)$, in subdividing each edge with two new degree-2 vertices, yielding a new graph G'. More precisely, each edge $e = uv$ is replaced by a a path $ux_{e,u}x_{e,v}v$, where $x_{e,u}$ ad $x_{e,v}$ are two new vertices. We claim the following: if OPT is the size of a minimum vertex cover of G, and if OPT' is the size of a minimum vertex cover of G', then $OPT' = OPT + |E|$. Suppose that C is a vertex cover of G. Let $C' = C \cup \{x_{e,u} : v \in C\} \cup \{x_{e,v} : v \notin C\}$. Then $|C'| = |C| + |E|$, and C' is a vertex cover of G'. Conversely, suppose that C is a minimum vertex cover of G'. We can assume that for every $e = uv$ in E, the set C contains only one of $x_{e,u}, x_{e,v}$ (for if both are in C, then $C \setminus \{x_{e,v}\} \cup \{v\}$ is a vertex cover of G' of the same size). It follows that for $e = uv$ in E: if $x_{e,u} \notin C$ then $u \in C$, and if $x_{e,v} \notin C$ then $v \in C$. Thus, the set $C' = C \cap V$ is a vertex cover of G of size $|C| - |E|$. $\square$

Proposition 2 *For all $l \geq 4$,* Annotated C_l-free edge deletion *has no polynomial kernel unless $NP \subseteq coNP/poly$.*

Proof We give a cross-composition from Vertex Cover in graphs of girth at least 9, which is NP-hard by Lemma 16. By choosing an appropriate polynomial equivalence relation, we can assume that we are given t instances $(G_1, k), ..., (G_t, k)$ of the problem, where all G_i's have the same number of vertices n. We assume that $t = 2^s$ (free to duplicate some instances if necessary). We let $p = \lfloor l/2 \rfloor + 1$. We shall construct an instance (G, F, k') of Annotated

C_l-FREE EDGE DELETION, with $k' = s + 2pn(n-1) + k$, and where G, F are constructed as follows. Non-allowed edges will be said as *forbidden*.

Construction. We first construct the graph T as in Claim 10. We start with a complete binary tree T_0 with root r and t leaves $s_1, \dots, s_t$; to each node u of T_0, we associate an edge e_u in T as follows: if u is associated to the edge xy and if u has two children v, v', we create a new vertex z and we let $e_v = xz, e_{v'} = yz$. All the edges of T are allowed. For all $1 \leq i \leq t$, we write e_i instead of e_{s_i}, and we denote by R_i the path in T from r to s_i.

Now, for each $1 \leq i \leq t$, we introduce in G three graphs G'_i, G''_i, P_i. Suppose that $G_i = (V_i, E_i)$. Let $r = \lfloor l/2 \rfloor - 1$ and $r' = \lceil l/2 \rceil - 1$, so that $r + r' = l - 2$. The graph G'_i is a copy of a r-subdivision of G_i, i.e. for each edge uv in G it contains a path p'_{uv} of length $r+1$ joining u' and v'. Likewise, the graph G''_i is a copy of a r'-subdivision of G_i, i.e. for each edge uv in G it contains a path p''_{uv} of length $r'+1$ joining u'' and v''. We mark the edges of G'_i, G''_i as forbidden. We add all edges between V'_i, V''_i and we mark these edges as allowed. Let S_i be the set of edges $u'v''$ $(u, v \in V_i)$ with $u \neq v$. For each edge $e \in S_i$, we add a "propagation gadget" $P_{i,e}$ such that the deletion of e_i entails the deletion of e. Suppose that $e_i = uv$ and that $e = xy$. The graph $P_{i,e}$ consists of $2p-2$ vertices $x_1, ..., x_{2p-2}$, and of $2p$ triangles uvx_1, vx_1x_2, $x_jx_{j+1}x_{j+2}$ $(1 \leq j \leq 2p-2)$, $x_{2p-1}x_{2p}x, x_{2p}xy$. We mark the edges vx_1, x_jx_{j+1} $(1 \leq j \leq 2p-3)$, $x_{2p-2}x$ as allowed, and the other edges as forbidden. We let P_i be the union of the graphs $P_{i,e}$ for $e \in S_i$.

To complete the construction, we do the following. First, for every allowed edge e of $T \cup \bigcup_{i=1}^{t} P_i$, we add a path Q_e of length $l-2$ joining the two endpoints of e. We mark the edges of these paths as forbidden. Finally, we remove the edge e_r. We let $V' = \cup_{i=1}^{t} V'_i, V'' = \cup_{i=1}^{t} V''_i$, P denote the union of the graphs P_i, and Q denote the union of the graphs Q_e.

Correctness. Suppose that (G, F, k') is a positive instance of ANNOTATED C_l-FREE EDGE DELETION, admitting a solution S. Let v and w be the two children of r in T_0. In T, the absence of the edge e_r implies that $Q_{e_u}e_ve_w$ form a C_l in G. Hence one of e_v, e_w must be in S. By iterating this process, there is $i \in [t]$ such that all the s edges associated to vertices in R_i are in S. In particular, $e_i \in S$. Now, for each edge e in S_i, the gadget $P_{i,e}$ implies that the allowed edges of $P_{i,e}$, as well as e, are in S. Let S' be the set of vertices $u \in V_i$ such that $u'u'' \in S$. We then have $s + 2pn(n-1) + |S'| \leq |S| \leq k'$, and thus $|S'| \leq k' - s - 2pn(n-1) = k$. We claim that S' is a vertex cover of G_i. Indeed, for each edge uv in G_i, since $u'p'_{uv}v'v''p''_{vu}u''$ cannot be a C_l in $G_i \setminus S$ it follows that one of $u'u'', v'v''$ must be in S, and thus one of u, v is in S'. We conclude that (G_i, k) is a positive instance of VERTEX COVER.

Suppose that (G_i, k) is a positive instance of VERTEX COVER, admitting a solution S. Let S' be the set containing contains (i) the edges of T associated to vertices of R_i, (ii) the allowed edges of $P_{i,e}$ and the edge e, for each $e \in S_i$, (iii) the edge $u'u''$ for each $u \in S$. We then have $|S'| = s + 2pn(n-1) + |S| \leq s + 2pn(n-1) + k = k'$. The following claim will allow us to conclude that (G, F, k') is a positive instance of ANNOTATED C_l-FREE EDGE DELETION.

Claim 17 *$G \setminus S'$ is C_l-free.*

Proof. Suppose that C is an induced C_l in $G \setminus S'$. We first consider the case when C intersects Q. Then C contains a path Q_e for some e allowed edge of $T \cup P$. Since Q_e has length $l-2$, C contains two extra edges e', e'', and the edge e is in S'. Assume first that e is an allowed edge of T, then $e = e_u$ for some u node of T_0. We have $e' = e_v, e'' = e_w$ with either (i) v, w children of u in T_0, or (ii) u, w children of v in T_0. In case (i) one of e', e'' is in S'; in case (ii) e' is in S'. This contradicts the assumption that C is an induced cycle. Assume now that e is an allowed edge of $P_{j,f}$. Then one of e', e'' is an allowed edge which is in S' by definition, a contradiction.

The second case to consider is when C does not intersect Q, but intersects P. Let us assume that C intersects $P_{j,e}$ with $e \in S_j$. Let $e_j = uv$ and $e = xy$. Observe that $P_{j,e} \setminus S'$ consists of either a chain of triangles, or of two vertex-disjoint paths. Therefore, if C intersects $P_{j,e}$, it follows that C contains a path in $P_{j,e}$ joining $\{u, v\}$ to $\{x, y\}$, and that this path has at least p edges. Since the vertices of P_j disconnect T from $V'_j \cup V''_j$, C must contain another path in some $P_{j,f}$, with at least p edges. We conclude that C has at least $2p > l$ edges, a contradiction.

The third case to consider is when C does not intersect P, Q, but intersects V' or V''. As P_j separates T from $V'_j \cup V''_j$, it follows that C is included in $V'_j \cup V''_j$ for some j. We cannot have G included in V'_j, as G'_j has girth $9r \geq l$. Likewise, we cannot have G included in V''_j. If $|C \cap V'_j| = 1$, then C has the form $u''p''_{u,v}v''p''_{v,w}w''x'$, and the edges $x'u'', x'w''$ imply that $j \neq i$ and thus the edge $x'v''$ is present, contradiction. We reason similarly if $|C \cap V''_j| = 1$. Suppose now that $|C \cap V'_j| \geq 2$ and that $|C \cap V''_j| \geq 2$. Then C has the form $u'p'_{u,v}v'y''p''_{y,x}x''$. The absence of the edges $u'y'', v'x''$ implies that $j = i$, and the presence of the edges $u'x'', v'y''$ implies that $u = x, v = y$. We obtain that uv is in G and that $u, v \notin S$, contradicting the assumption that S is a vertex cover of G.

The last case to consider is when C does not intersect V', V'', P, Q. Then C is included in T. We claim that $T \setminus S'$ is a chordal graph. Indeed, it can be obtained in the following way: when considering a node u with children v, w, such that $e_u = xy$, then (i) if $e_u \neq e_r$ and $e_u \notin S'$ then add a new vertex z adjacent to x, y, and let $e_v = xz, e_w = yz$, (ii) otherwise, one of e_v, e_w is in S'; if $e_v \in S'$ then add a new vertex z adjacent to y, and let $e_w = yz$; proceed similarly if $e_w \in S'$. Since $T \setminus S'$ is a chordal graph, it cannot contain an induced C_l with $l \geq 4$, contradiction. □

We are now in a position to prove the following:

Theorem 8 *The C_l-FREE EDGE DELETION problem has no polynomial kernel for any $l \geq 4$, unless $NP \subseteq coNP/poly$.*

Proof We give a polynomial-time-and-parameter transformation from ANNOTATED C_l-FREE EDGE DELETION. Let (G, F, k) be an instance of ANNOTATED C_l-FREE EDGE DELETION, where $G = (V_G, E_G)$. We construct an instance

(H, k) of C_l-FREE EDGE DELETION as follows. The graph $H = (V_H, E_H)$ is obtained from G by adding, for each edge $e = uv$ in F, a set V_e of vertices consisting of : (i) $k+1$ vertices $x_{0,e}, ..., x_{k,e}$, (ii) $k+1$ paths of length $l-3$ $p_{0,e}, ..., p_{k,e}$. Let $y_{i,e}, z_{i,e}$ be the two endpoints of $p_{i,e}$, then: (i) we make $x_{i,e}, y_{i,e}$ adjacent to u, (ii) we make $x_{i,e}, z_{i,e}$ adjacent to v.

Let us prove the correctness of the transformation. Suppose that there exists a set S of allowed edges of size at most k such that $G' = (V_G, E_G \backslash S)$ is C_l-free. We show that $H' = (V_H, E_H \backslash S)$ is C_l-free. Suppose by contradiction that C is an induced C_l in H'. As C cannot be an induced C_l in G', it has to intersect some set V_e with $e = uv$ in F. If C contains a vertex $x_{i,e}$, its neighbors in C are the vertices u, v yielding an induced C_3, impossible. If C contains a path $p_{i,e}$, then in C the vertex $y_{i,e}$ is adjacent to u and the vertex $z_{i,e}$ is adjacent to v, which yields an induced C_{l-1}, impossible. Conversely, suppose that there exists a set S of at most k edges such that $H' = (V_H, E_H \backslash S)$ is C_l-free. It suffices to show that S is disjoint from F, as this implies that $S' = S \cap E_G$ is a set of allowed edges of size at most k such that $G' = (V_G, E_G \backslash S')$ is C_l-free. Suppose by contradiction that S contains an edge $e = uv$ in F. As $|S| \leq k$, we can find an i such that the path $ux_{i,e}v$ is present in H', and we can find a j such that the path $up_{j,e}v$ is present in H'. Since uv is not an edge of H', these two paths form a C_l in H', contradiction. □

5 Conclusion

In this paper, we provide evidence that the C_l-FREE EDGE-DELETION and the P_l-FREE EDGE-DELETION problems do not admit polynomial kernels for large enough l (unless $NP \subseteq coNP/poly$ [1]). These problems were left open by Kratsch and Wahlström in [18]. While our result for C_t-FREE EDGE-DELETION is best possible, it remains open whether the P_t-FREE EDGE-DELETION problem admits a polynomial kernel for $4 \leqslant t < 7$.

Moreover, we have shown that the PARAMETERIZED COGRAPH EDGE MODIFICATION problems admit vertex cubic kernels. It would be interesting to really determine why these results hold. There are few possible reasons: the first is the P_4-free characterization of cographs, the second is the property of being totally decomposable with respect to the modular decomposition. Because of the negative results for P_t-free graphs with $t \geqslant 7$, we suspect the forbidden subgraph characterization is not enough. To push further the idea that having a nice tree-decomposition scheme is important, we should investigate whether other decompositions can be used to achieve polynomial kernels for edge-modification problems. An interesting candidate would be the split decomposition [7,19] which provides a decomposition similar to the cotree for distance hereditary graphs. Moreover, it is known that distance hereditary graphs do not have a finite set of forbidden induced subgraphs.

To conclude, we mention that cographs are exactly clique-width 2 graphs [6] and that distance hereditary graph are exactly rank-width one graphs [23]. What about kernelization for edge-modification problems for small value of such classical width-parameters?

Acknowledgment. Research supported by the AGAPE project (ANR-09-BLAN-0159).

References

1. H. Bodlaender, R. Downey, M. Fellows, and D. Hermelin. On problems without polynomial kernels. In *ICALP*, number 5125 in LNCS, pages 563–574, 2008.
2. H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Cross-composition: A new technique for kernelization lower bounds. In *STACS*, volume 9 of *LIPIcs*, pages 165–176, 2011.
3. H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *ESA*, volume 5757 of *LNCS*, pages 635–646, 2009.
4. A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey.* SIAM Monographs on Discrete Mathematics and Applications. 1999.
5. L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.
6. B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle rewriting hypergraph grammars. *Journal of Computing System Science*, 46:218–270, 1993.
7. W. Cunningham and J. Edmonds. A combinatorial decomposition theory. *Canadian Journal of Mathematics*, 32(3):734–765, 1980.
8. R. Downey and M. Fellows. *Parameterized complexity.* Springer, 1999.
9. E. S. El-Mallah and C. Colbourn. The complexity of some edge deletion problems. *IEEE Transactions on Circuits and Systems*, 35(3):354–362, 1988.
10. M. Fellows, M. Langston, F. Rosamond, and P. Shaw. Efficient parameterized preprocessing for cluster editing. In *Fundamentals of Computation Theory, 16th International Symposium, (FCT)*, number 4639 in LNCS, pages 312–321, 2007.
11. J. Flum and M. Grohe. *Parameterized complexity theorey.* Texts in Theoretical Computer Science. Springer, 2006.
12. L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *STOC*, pages 133–142, 2008.
13. M. Garey and S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness.* Freeman, 1978.
14. M. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. *Journal of Algorithms*, 19:449–473, 1995.
15. M. Habib and C. Paul. A survey on algorithmic aspects of modular decomposition. *Computer Science Review*, 4(1):41–59, 2010.
16. P. Heggernes, C. Paul, J. A. Telle, and Y. Villanger. Interval completion with few edges. In *STOC*, pages 374–381, 2007.
17. C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 95–103, 2007.
18. S. Kratsch and M. Wahlström. Two edge modification problems without polynomial kernels. In *IWPEC*, volume 5917 of *LNCS*, pages 264–275, 2009.
19. T.-H. Ma and J. Spinrad. An $O(n^2)$ algorithm for undirected split decomposition. *Journal of Algorithms*, 16:145–160, 1994.
20. A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113(1):109 – 128, 2001.
21. R. Niedermeier. *Invitation to fixed parameter algorithms*, volume 31 of *Oxford Lectures Series in Mathematics and its Applications.* Oxford University Press, 2006.
22. R. Niedermeier and P. Rossmanith. A general method to speed up fixed-parameter-tractable algorithms. *Information Processing Letters*, 73(3-4):125–129, 2000.
23. S. Oum. *Graphs of bounded rank-width.* PhD thesis, Princeton University, 2005.
24. D. Rose. A graph-theoretic study of the numerical solution of sparse positive systems of linear equations. *Graph Theory and Computing*, pages 183–217, 1972.
25. R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.

26. R. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM journal of Computing*, 13:566–579, 1984.
27. A. van Zuylen and D. Williamson. Deterministic algorithms for rank aggragation and other ranking and clustering problems. In *WAOA*, volume 4927 of *LNCS*, pages 260–273, 2008.