Incompressibility of H-Free Edge Modification*

Leizhen Cai** and Yufei Cai***

Department of Computer Science and Engineering,
The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China
lcai@cse.cuhk.edu.hk, cai@mathematik.uni-marburg.de

Abstract. Given a fixed graph H, the H-Free Edge Deletion (resp., Completion, Editing) problems ask whether it is possible to delete from (resp., add to, delete from or add to) the input graph at most k edges so that the resulting graph is H-free, i.e., contains no induced subgraph isomorphic to H. These H-free edge modification problems are well known to be FPT for every fixed H. In this paper, we study the nonexistence of polynomial kernels for them in terms of the structure of H, and completely characterize their nonexistence for H being paths, cycles or 3-connected graphs. As a very effective tool, we have introduced a constrained satisfiability problem Propagational Satisfiability to cope with the propagation of edge additions/deletions, and we expect the problem to be useful in studying the nonexistence of polynomial kernels.

1 Introduction

Edge modification problems are concerned with adding edges to or deleting edges from input graphs to obtain graphs with desired properties, and have been studied extensively under frameworks of both traditional complexity and parameterized complexity. In this paper, we focus on edge modification problems concerning the property of being *H-free* for a fixed graph H, i.e., our desired graph contains no induced subgraph isomorphic to H. Such problems are fundamental as any hereditary property is H-free for every graph H in a set of forbidden induced subgraphs. We consider the following H-free edge modification problems.

H-Free Edge Deletion
Instance: Graph G, and *parameter* k.
Question: Can we delete from G at most k edges to make it H-free?

H-Free Edge Completion and H-Free Edge Editing are defined similarly by replacing "delete from" with "add to" and "delete from or add to" respectively.

* Partially based on the MPhil Thesis of the 2nd author under the supervision of the 1st author.

** Partially supported by GRF grant CUHK410409 of the Research Grants Council of Hong Kong.

*** Current address: Philipps-Universitaet, Marburg Mehrzweckgebaeude 05D06, Hans-Meerwein Straβe, 35032 Marburg, Gemany.

G. Gutin and S. Szeider (Eds.): IPEC 2013, LNCS 8246, pp. 84–96, 2013.
© Springer International Publishing Switzerland 2013

The above H-free edge modification problems are FPT for every fixed H following a general result of the first author [6]. In IWPEC'06 [2], the same author raised the issue of determining the existence of polynomial kernels for H-Free Edge Deletion in terms of the structure of H. Kratsch and Wahlström [11] constructed the first H for which neither H-Free Edge Deletion nor H-Free Edge Editing admits polynomial kernels, and Guillemot et al. [10] established the nonexistence of polynomial kernels for H-Free Edge Deletion when H is a path P_l with $l \geq 13$ or a cycle C_l with $l \geq 12$, provided that coNP $\not\subseteq$ NP/poly. On the other hand, Gramm et al. [9] obtained polynomial kernels for P_3-Free Edge Deletion, Completion and Editing, and Guillemot et al. [10] presented polynomial kernels for P_4-Free Edge Deletion, Completion and Editing. Other than the above results, very little was known regarding polynomial kernels of H-free edge modification problems.

In this paper, we study the nonexistence of polynomial kernels for H-free edge modification problems in terms of the structure of H. We fully characterize 3-connected H for which H-free edge modification problems admit no polynomial kernel, and determine exactly when P_l- or C_l-free edge modification problems admit no polynomial kernel, assuming coNP $\not\subseteq$ NP/poly.

- For 3-connected H, H-Free Edge Deletion and Editing admit no polynomial kernel iff H is not a complete graph.
- For 3-connected H, H-Free Edge Completion admits no polynomial kernel iff H misses at least two edges.
- For H being a path or cycle, H-Free Edge Deletion, Completion and Editing admit no polynomial kernel iff H has at least 4 edges.

We assume that the reader is familiar with the general framework for kernelization lower bounds [1, 3–5, 8]. In the paper, our kernels refer to *generalized kernels* [4] (called *bikernels* by Alon et al. [1]).

Definition 1. [1, 4] *A* generalized kernelization *from a parameterized problem Π into another parameterized problem Π' is an algorithm that takes any instance $(I, k) \in \Pi$ as input, runs in time polynomial in $|I| + k$, and outputs an instance $(I', k') \in \Pi'$ such that*

(a) *(I, k) is a yes-instance of Π iff (I', k') is a yes-instance of Π', and*
(b) *both $|I'|$ and k' are bounded by a function $g(k)$ on k alone.*

The output (I', k') is called a generalized kernel*, and it is a* polynomial kernel *if $g(k)$ is a polynomial.*

A *polynomial parameter transformation* (Bodlaender et al. [5]) from a parameterized problem Π into another parameterized problem Π' is the same as a generalized kernelization with condition (b) changed to "*the value of parameter k' is bounded by a polynomial of k*". For simplicity, we call a parameterized problem *incompressible* if it has no polynomial kernel unless coNP $\not\subseteq$ NP/poly.

To obtain our results, first we introduce a constrained satisfiability problem Propagational Satisfiability and prove its incompressibility (Section 2).

Then we use it as our seed problem for polynomial parameter transformations to establish the incompressibility of some "quarantined" H-free edge modification problems where we have a restriction on edges that can be added/deleted (Section 3). Finally we lift the quarantine by using "enforcers" (Section 4), and discuss some open problems (Section 5).

Our results significantly improve our knowledge on the incompressibility of H-free edge modification problems, and our PROPAGATIONAL SATISFIABILITY problem is very useful in coping with the propagation of edge deletions/additions and thus the incompressibility of edge modification problems. We hope that our ideas will be useful in the discovery of a dichotomy theorem on the incompressibility of H-free edge modification problems, and we also expect PROPAGATIONAL SATISFIABILITY to be useful in studying the nonexistence of polynomial kernels in general.

2 Satisfiability of Propagational Formulas

One main complication of H-free edge modification problems lies in the possibility of introducing new induced copies of H when we add/delete edges, which causes a propagation of edge additions/deletions. To cope with this, we introduce in this section a constrained satisfaction problem PROPAGATIONAL SATISFIABILITY and establish its incompressibility, and we will use the problem extensively to show the incompressibility of our edge modification problems.

Definition 2. *A ternary Boolean function $f(x, y, z)$, where x, y and z are either Boolean variables or constants 0 or 1, is* propagational *if it satisfies $f(1, 0, 0) = 0$ and $f(0, 0, 0) = f(1, 0, 1) = f(1, 1, 0) = f(1, 1, 1) = 1$.*

In other words, $f(x, y, z)$ is propagational if it is true when either $x = y = z = 0$ or "$x = 1$ implies $y = 1$ or $z = 1$". There are eight different propagational functions f in total due to the freedom of defining the value of f for the other three assignments of variables.

Example 3. *The following three functions are propagational:*

$$
\begin{aligned}
f_1(x, y, z) &= \overline{x} \vee y \vee z, \\
f_2(x, y, z) &= x \,\texttt{XOR}\, (y \,\texttt{NOR}\, z), \\
\texttt{Not-1-in-3}(x, y, z) &= (\overline{x} \vee y \vee z) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee y \vee \overline{z}).
\end{aligned}
$$

Propagational functions $f(x, y, z)$ generalize function `Not-1-in-3` of Kratsch and Wahlström [11], and capture the relation that "whatever happens to x must happen to either y or z", which is of great use when we deal with edge modification problems because of propagations of edge deletions/additions. The following example of C_4-FREE EDGE DELETION explains such a connection. Suppose that we want to delete some light edges from the graph in Fig. 1 to obtain a C_4-free graph. When we delete edge x, we create a new induced C_4 in the graph, and we must delete either edge y or edge z or both in order to destroy the new C_4.

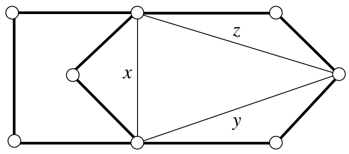

Fig. 1. Realization of a propagational function $f(x, y, z)$ by C_4-free edge deletion

Therefore we can use the graph to realize a propagational function $f(x, y, z)$, which also represents the propagation of edge deletions from x to y or z.

For a Boolean function $f(x, y, z)$, a *conjunctive formula* φ is of the form

$$f(x_1, y_1, z_1) \wedge f(x_2, y_2, z_2) \wedge \cdots \wedge f(x_m, y_m, z_m).$$

Each $f(x_i, y_i, z_i)$ is a *clause* of φ, and the *Hamming weight* of an assignment of 0's and 1's to variables is the number of 1's in the assignment. For φ, the *degree of a variable* is its number of occurrences in φ, and the *degree of* φ is the maximum degree of its variables. We say that φ is *t-regular* if all its variables have degree t.

PROPAGATIONAL SATISFIABILITY
Instance: Conjunctive formula φ of a propagational function f, *parameter* k.
Question: Does φ have a satisfying assignment of Hamming weight $\leq k$?

We will establish the incompressibility of the above problem in two steps: first prove its NP-completeness, and then show that it is OR-compositional.

Lemma 4. *For any propagational function $f(x, y, z)$,* PROPAGATIONAL SATISFIABILITY *is NP-complete on degree-6 conjunctive formulas with one occurrence of constant 1.*

Proof. The problem is clearly in NP, and we give a polynomial reduction from the classical VERTEX COVER problem. For an arbitrary instance (G, k) of VERTEX COVER, we first construct a conjunctive formula φ': each vertex of G is a variable and each edge uv of G corresponds to a clause $f(1, u, v)$, which forces us to choose either u or v in order to satisfy $f(1, u, v)$. Clearly G has a vertex cover of size $\leq k$ iff φ' can be satisfied with $\leq k$ true variables.

Next we convert φ' into a degree-6 conjunctive formula φ with one occurrence of constant 1. Note that two clauses $f(x, y, 0)$ and $f(y, x, 0)$ ensure that x and y have the same value, and we write $(x = y)$ as a short hand for $f(x, y, 0) \wedge f(y, x, 0)$. Given any $p = 2^q$, we can make variables $w_1, \ldots, w_{2p-1}$ take the same value by adding the following set $F(w)$ of clauses

$$\begin{aligned}
&(w_1 = w_2) \wedge (w_1 = w_3) \wedge \\
&(w_2 = w_4) \wedge (w_2 = w_5) \wedge (w_3 = w_6) \wedge (w_3 = w_7) \wedge \\
&\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\
&(w_{p/2} = w_p) \wedge (w_{p/2} = w_{p+1}) \wedge \cdots \wedge (w_{p-1} = w_{2p-2}) \wedge (w_{p-1} = w_{2p-1}).
\end{aligned}$$

Among these clauses, w_1 appears four times, $w_p, \ldots, w_{2p-1}$ appear twice, and other variables appear six times (recall that $(w_i = w_j)$ means two appearances of w_i and w_j each). The variables $w_1, w_p, \ldots, w_{2p-1}$ can be used in other clauses.

Let m be the number of edges of G, and choose $p = 2^q$ between $3m$ and $6m-1$. We construct from φ' a degree-6 formula φ with one occurrence of constant 1.

1. Add a variable w_1 not occurring in φ' and the clause $f(1, w_1, 0)$, which forces w_1 to take value 1.
2. Add variables $w_2, \ldots, w_{2p-1}$ not occurring in φ' to represent occurrences of 1 in φ', and add clauses $F(w)$ to force all $w_2, \ldots, w_{2p-1}$ to take value 1.
3. For every variable v of φ', add variables $v_1, \ldots, v_{2p-1}$ and clauses $F(v)$ to force all $v_1, \ldots, v_{2p-1}$ to have the same value.
4. For the i-th clause $f(1, u, v)$ of φ', add clause $f(w_{p+3i-3}, u_{p+3i-2}, v_{p+3i-1})$. Since $i \leq m$ and $3m \leq p$, we will never run out of variables.

If φ' is satisfiable with $\leq k$ true variables, we can satisfy φ with $(k+1)(2p-1) \leq 12(k+1)m$ true variables, consisting of $w_1, \ldots, w_{2p-1}$ and those $v_1, \ldots, v_{2p-1}$ for $v = 1$ in the satisfying assignment to φ'. The converse is also true. □

Lemma 5. *For any propagational function $f(x, y, z)$,* Propagational Satisfiability *is OR-compositional on degree-6 conjunctive formulas with one occurrence of constant 1.*

Proof. We describe a composition algorithm very similar to the one in Lemma 2 of Kratsch and Wahlström [11]. Let $(\varphi_1, k), \ldots, (\varphi_t, k)$ be t instances of the problem such that each φ_i has degree 6 and one occurrence of 1. Note that each φ_i can be solved in $O(3^k|\varphi_i|)$ time by bounded search tree. If $t > 2^k$, we have enough time to solve each φ_i and output a dummy yes- or no-instance accordingly.

Therefore we assume $t \leq 2^k$, and let $p = 2^q$ be the power of two between t and $2t-1$. Construct a conjunctive formula φ' as follows.

1. Rename variables of $\varphi_1, \ldots, \varphi_t$ so that they are all distinct.
2. Add all clauses of $\varphi_1, \ldots, \varphi_t$ to φ'.
3. For each φ_i, replace the occurrence of 1 with a distinct variable w_{i+p-1}.
4. Add the following clauses so that $\geq q$ variables from $w_2, \ldots, w_{2p-1}$, including one of $w_p, \ldots, w_{2p-1}$, are forced to take value 1:

$$\begin{array}{lll} f(1, w_2, w_3)\wedge & & \\ f(w_2, w_4, w_5) \wedge f(w_3, w_6, w_7)\wedge & & \\ \qquad\vdots & & \vdots \\ f(w_{p/2}, w_p, w_{p+1}) \wedge \cdots \wedge f(w_{p-1}, w_{2p-2}, w_{2p-1}). & & \end{array}$$

Set $k' = k + q \leq 2k$. If some (φ_i, k) is a yes-instance, then we can satisfy φ' with q true variables from $w_2, \ldots, w_{2p-1}$ including w_{i+p-1} and $\leq k$ additional true variables satisfying clauses of φ_i, for a total of k' true variables. The clauses of

φ_j with $j \neq i$ are satisfied with 0 in all 3 positions. Conversely, if (φ', k') is a yes-instance, then $\geq q$ variables from $w_2, \ldots, w_{2p-1}$ are forced to be true, including some w_{i+p-1}. Then clauses of φ_i are satisfied with the remaining quota of $\leq k$ true variables. □

Theorem 6. *For any propagational function $f(x, y, z)$,* Propagational Satisfiability *on 6-regular conjunctive formulas is incompressible.*

Proof. By Lemma 4 and Lemma 5 and the work in [4], Propagational Satisfiability is incompressible on degree-6 conjunctive formulas with one occurrence of 1. We can easily modify a degree-6 conjunctive formula into an equivalent 6-regular conjunctive formula: For each variable x of degree d, add $(6 - d)$ clauses of the form $f(1, 1, x)$. □

3 Incompressibility: Quarantined Edge Modification

To ease the complication in tackling H-free modification problems, we first add a restriction to edges that can be added or deleted, which forms "quarantined" edge modification problems. We then use our incompressible propagational satisfiability problems to show that "quarantined" edge modification problems are incompressible for H being a 4-cycle, 5 cycle, or 3-connected graph, which forms the base for our main results.

Quarantined H-Free Edge Deletion
Instance: Graph G, forbidden set $F \subseteq E(G)$, and *parameter* k.
Question: Can we delete at most k edges from $E(G) - F$ to make G H-free?

Edges in F are *forbidden edges*, edges in $E(G) - F$ are *allowed edges*, and allowed edges form the *allowed subgraph*.

Quarantined H-Free Edge Completion.
Instance: Graph G, forbidden set $F \subseteq E(\overline{G})$, and *parameter* k.
Question: Can we add at most k edges from $E(\overline{G}) - F$ to make G H-free?

Note that $\overline{G}$ is the complement of G. Edges in F are *forbidden nonedges*, edges in $E(\overline{G}) - F$ are *allowed nonedges*, and allowed nonedges form the *allowed complement* of G.

Theorem 7. Quarantined C_4-Free Edge Deletion *is incompressible on graphs whose allowed subgraphs contain no C_4 as a partial subgraph.*

Proof. We give a polynomial parameter transformation from Propagational Satisfiability on 6-regular conjunctive formulas of propagational function `Not-1-in-3`. We need the three components in Fig. 2, where an edge marked with a letter, say x, will be referred to as an *x-edge*.

For an arbitrary instance (φ, k) of our Propagational Satisfiability, we construct an instance (G, F, k') of Quarantined C_4-Free Edge Deletion as follows (see Fig. 3 for an example).

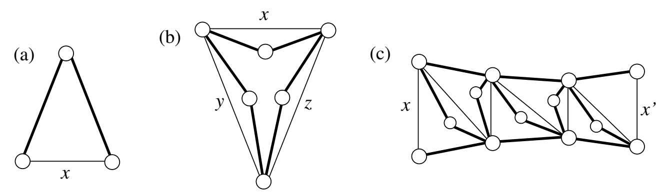

Fig. 2. Components for QUARANTINED C_4-FREE EDGE DELETION with thick edges indicating forbidden edges in F: (a) truth-setting component T(x), (b) satisfaction-testing component $S(x,y,z)$, and (c) communication component $C(x)$. Note that $S(x,y,z)$ realizes `Not-1-in-3`(x,y,z) as $S(x,y,z)$ itself is C_4-free and we need to delete at least two edges from $\{x,y,z\}$ to ensure that $S(x,y,z)$ stays C_4-free.

1. Create a truth-setting component $T(x)$ for each variable x of φ, and a satisfaction-testing component $S(x,y,z)$ for each clause $f(x,y,z)$ of φ.
2. For each clause $f(x,y,z)$, consider each $v \in \{x,y,z\}$. If $v \in \{0,1\}$, then the v-edge in $S(x,y,z)$ is deleted if $v = 1$ and marked as forbidden if $v = 0$. Otherwise v is a variable, and we add a communication component $C(v)$, identify the v-edge of $T(v)$ with the v-edge of $C(v)$ and identify the v'-edge of $C(v)$ with the v-edge of $S(x,y,z)$.
3. Let G be the resultant graph, F the set of forbidden edges in all components, and set $k' = 37k$.

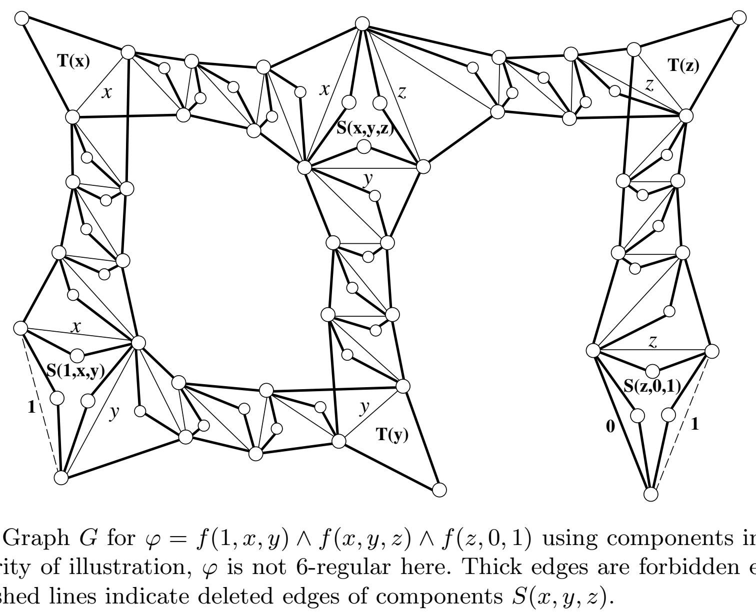

Fig. 3. Graph G for $\varphi = f(1,x,y) \wedge f(x,y,z) \wedge f(z,0,1)$ using components in Fig. 2. For clarity of illustration, φ is not 6-regular here. Thick edges are forbidden edges F, and dashed lines indicate deleted edges of components $S(x,y,z)$.

It is easy to see that the allowed subgraph of G contains no C_4 as a partial subgraph, and the transformation is a polynomial parameter transformation. We show that φ is satisfiable with $\leq k$ true variables iff G can be made C_4-free by deleting $\leq k'$ allowed edges.

($\Rightarrow$) Consider a satisfying assignment of φ with $\leq k$ true variables, and let E' be allowed edges of all copies of communication components in G for all true variables. Each variable x has 6 communication components with one shared x-edge and contributes 37 edges to E', implying $|E'| \leq 37k = k'$. It is easily checked that $G - E'$ is C_4-free.

($\Leftarrow$) Let E' be a set of $\leq k'$ allowed edges in G whose deletion results in a C_4-free graph. Observe that for a communication component $C(x)$ in G, as far as C_4-freeness is concerned, deleting its x-edge will force the deletion of all its allowed edges, including x'-edge. It follows that for every truth-setting component $T(x)$, the 37 allowed edges of communication components attached to $T(x)$ are either all deleted or none deleted. Assign $x = 1$ if the x-edge of $T(x)$ is in E' and assign $x = 0$ otherwise, and we have set $\leq k$ variables true. For each clause $f(x, y, z)$, it is ensured by the C_4-freeness of its satisfaction-testing component $S(x, y, z)$ after deleting E' that $f(x, y, z)$ satisfies `Not-1-in-3` and thus is true. □

With the above theorem, we can easily give a polynomial parameter transformation from Quarantined C_4-Free Edge Deletion to $\overline{P_5}$-Free Edge Deletion, where $\overline{P_5}$ is the same as the house graph $C_5 + e$ [7].

Corollary 8. $\overline{P_5}$-Free Edge Deletion *is incompressible.*

The construction and proof in Theorem 7 highlight the basic ideas in establishing the incompressibility of Quarantined H-Free Edge Deletion and Completion:

1. Use $T(x)$ to decide whether to assign 0 or 1 to x.
2. Use $S(x, y, z)$ to realize a propagational function f.
3. Use $C(x)$ to represent the propagation of edge deletions/additions from x-edges to x'-edges, and connect $T(x)$ with satisfaction-testing components.

Indeed, we can establish the incompressibility of Quarantined C_4-Free Edge Completion in a way almost identical to the proof of Theorem 7: use the components in Fig. 4, instead of those in Fig. 2. We also use a different propagational function $f(x, y, z) = x \,\texttt{XOR}\, (y \,\texttt{NOR}\, z)$, instead of `Not-1-in-3`.

Theorem 9. Quarantined C_4-Free Edge Completion *is incompressible on graphs whose allowed complements have girth greater than* 4.

As in the case of $\overline{P_5}$-free edge deletion, we can use Theorem 9 to construct an easy polynomial parameter transformation for the incompressibility of $\overline{P_5}$-free edge completion [7].

Corollary 10. Quarantined $\overline{P_5}$-Free Edge Completion *is incompressible.*

Very similar constructions also work for C_5-Free Edge Deletion and Completion. Here we only give the key components, satisfaction-testing components

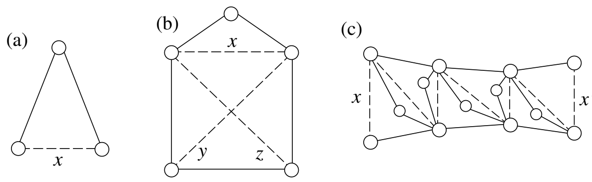

Fig. 4. Components for Quarantined C_4-Free Edge Completion (where allowed nonedges are denoted by dashed lines and forbidden nonedges are invisible): (a) truth-setting component T(x), (b) satisfaction-testing component $S(x, y, z)$, and (c) communication component $C(x)$. Note that $S(x, y, z)$ realizes $f(x, y, z) = x\,\texttt{XOR}\,(y\,\texttt{NOR}\,z)$: when we add some edges in $\{x, y, z\}$, the resulting graph is C_4-free iff the string xyz is 000, 101, 110 or 111.

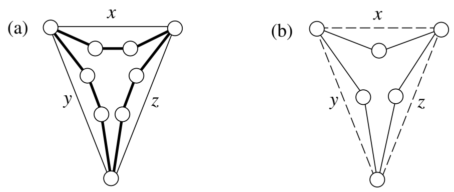

Fig. 5. Satisfaction-testing components $S(x, y, z)$ for (a) C_5-Free Edge Deletion and (b) C_5-Free Edge Completion. Both realize propagational function `Not-1-in-3`.

$S(x, y, z)$, in Fig. 5, and full proofs are available in [7] where it describes a general scheme for this type of constructions.

We now turn to 3-connected H for which similar constructions also work. Here, again we only give the construction of satisfaction-testing components $S(x, y, z)$, and full proofs are available in [7] and will be given in the full paper.

Quarantined H-Free Edge Deletion *for H being 3-connected but not complete.* H contains an induced $P_3 = a, b, c$. Let x be the nonedge ac, y edge ab and z edge bc; and we set $S(x, y, z)$ to $H + x$. Regard edges x, y and z as Boolean variables. For an edge $e \in \{x, y, z\}$, assign to it value 1 iff it is deleted from $S(x, y, z)$, and define Boolean function $f(x, y, z) = 1$ iff the graph obtained from $S(x, y, z)$ is H-free when we delete from $S(x, y, z)$ edges in $\{x, y, z\}$ with value 1. It is easily checked that $f(0, 0, 0) = f(1, 0, 1) = f(1, 1, 0) = f(1, 1, 1) = 1$ but $f(1, 0, 0) = 0$, implying that f is propagational.

Quarantined H-Free Edge Completion *for H being 3-connected with at least 2 nonedges.* Let x be an arbitrary edge and y, z two nonedges of H, and we delete edge x from H to form $S(x, y, z)$. For an edge $e \in \{x, y, z\}$, assign to it value 1 iff it is added to $S(x, y, z)$, and define Boolean function $f(x, y, z) = 1$ iff the graph obtained from $S(x, y, z)$ is H-free when we add to $S(x, y, z)$ edges in $\{x, y, z\}$ with value 1. Again, $f(0, 0, 0) = f(1, 0, 1) = f(1, 1, 0) = f(1, 1, 1) = 1$ but $f(1, 0, 0) = 0$, and we get a propagational f.

We now summarize our results of the section in the following theorem, which will be used in the next section to obtain our main results.

Theorem 11. Quarantined H-Free Edge Deletion *is incompressible if* H *is* C_4*,* C_5*,* $\overline{P_5}$*, or a* 3*-connected graph with at least one nonedge.* Quarantined H-Free Edge Completion *is incompressible if* H *is* C_4*,* C_5*,* $\overline{P_5}$*, or a* 3*-connected graph with at least two nonedges.*

4 Lifting the Quarantine

In the previous section, we have shown the incompressibility of quarantined H-free edge deletion/completion problems. We now discuss how to lift the quarantine so that our results extends to our original unquarantined H-free edge deletion/completion problems. Furthermore, our tools will also allow us to easily extend incompressibility of edge deletion/completion to edge editing.

We need a way to prevent an edge from being deleted for edge deletion and a nonedge from being added for edge completion. This is in fact pretty straightforward: for each forbidden edge $e \in F$ attach $k+1$ vertex-disjoint copies of $H + e'$ (where e' is any nonedge of H) by identifying e' with e, and for each forbidden nonedge $e \in F$ attach $k+1$ vertex-disjoint copies of $H - e'$ (where e' is any edge of H) by identifying e' with e. *The trick is to prevent the introduction of new induced* H *in the process.*

Definition 12. *An* H-free deletion enforcer *is an* H*-free graph* H' *with a distinguished edge* e' *such that* (*a*) $H' - e'$ *has an induced* H*, and* (*b*) *the identification of* e' *with any edge* e *of a vertex-disjoint* H*-free graph produces an* H*-free graph.*

Definition 13. *An* H-free completion enforcer *is an* H*-free graph* H' *with a distinguished nonedge* e' *such that* (*a*) $H' + e'$ *has an induced* H*, and* (*b*) *the identification of* e' *with any nonedge* e *of a vertex-disjoint* H*-free graph produces an* H*-free graph.*

It is clear that the above notion of enforcers will enable us to lift quarantine by attaching $k+1$ enforcers to prevent an edge from being deleted or a nonedge from being added, without introducing unwanted copies of induced H. As a bonus, completion (resp., deletion) enforcers establish incompressibility of edge editing problems directly from that of edge deletion (resp., completion) problems: *by forbidding all nonedge with completion enforcers, editing is forced to be deletion; and by forbidding all edges with deletion enforcers, editing is forced to be completion.*

Lemma 14. H-Free Edge Deletion *(resp.* H-Free Edge Completion*) is incompressible if* Quarantined H-Free Edge Deletion *(resp.* Quarantined H-Free Edge Completion*) is incompressible and there exists an* H*-free deletion (resp. completion) enforcer.*

Lemma 15. *H-FREE EDGE EDITING is incompressible if either H-FREE EDGE DELETION is incompressible and there exists an H-free completion enforcer or H-FREE EDGE COMPLETION is incompressible and there exists an H-free deletion enforcer.*

The following claims can be easily verified.

1. For any $t \geq 4$, adding any chord e' to C_t yields a C_t-free deletion enforcer, and deleting any edge e' from C_t yields a C_t-free completion enforcer.
2. If H is 3-connected and has a nonedge e', then $H + e'$ is an H-free deletion enforcer.
3. If H is 3-connected then for any edge e', $H - e'$ is an H-free completion enforcer.

We also need the following easy to see but very useful facts.

Lemma 16. *H-FREE EDGE DELETION is equivalent to $\overline{H}$-FREE EDGE COMPLETION, and H-FREE EDGE EDITING is equivalent to $\overline{H}$-FREE EDGE EDITING.*

Now we are ready to state our characterization for 3-connected H.

Theorem 17. *Let H be 3-connected and assume* coNP $\not\subseteq$ NP/poly.

1. *H-FREE EDGE COMPLETION admits no polynomial kernel iff H has ≥ 2 nonedges.*
2. *H-FREE EDGE DELETION and H-FREE EDGE EDITING admit no polynomial kernel iff H is not a complete graph.*

Proof. Incompressibility follows from Theorem 11, the existence of enforcers, Lemma 14, and Lemma 15.

For the cases that admit polynomial kernels, H-FREE EDGE COMPLETION is trivial for H being a complete graph, and easily solved in $O(kn^t)$ for H being $K_t - e$ for some constant t (for each H found in G just add the missing edge): both solvable in polynomial time and thus have trivial kernels.

When H is a complete graph K_t for some constant t, K_t-FREE EDGE EDITING is equivalent to K_t-FREE EDGE DELETION. The latter admits a polynomial kernel by reducing it to HITTING SET where each subset has size $\binom{t}{2}$, and we can make the kernel an instance of K_t-FREE EDGE DELETION if one insists [7]. □

Theorem 18. *Let H be a path or cycle and assume* coNP $\not\subseteq$ NP/poly. *H-FREE EDGE DELETION, COMPLETION and EDITING have no polynomial kernel iff H has at least 4 edges.*

Proof. For $H = P_t$ with $t \leq 4$, polynomial kernels for these problems are found by Gramm et al. [9] and Guillemot et al. [10]. Since $C_3 = K_3$, polynomial kernels exist for these problems when $H = C_3$ as discussed in the proof of Theorem 17.

For the incompressibility part, if $H = C_t$ or P_t with $t \geq 6$ then $\overline{H}$ is 3-connected with at least two nonedges, and the incompressibility of these problems follow from Theorem 17, Lemma 14 and Lemma 16. For the remaining three cases $H = C_4, C_5$ or P_5, their incompressibility follow from Theorem 11, Lemma 14, Lemma 15, and Lemma 16. □

5 Conclusion: Towards a Dichotomy Theorem

Our incompressibility for 3-connected H is actually much more powerful than it looks in Theorem 17. Because of Lemma 16, Theorem 17 implies the following result which covers a very extensive range of H.

Corollary 19. *For any fixed H, H-FREE EDGE DELETION (resp. COMPLETION, and EDITING) is incompressible whenever H or $\overline{H}$ is 3-connected with at least two nonedges.*

From this we can deduce that for most trees H and for most disconnected H, H-free edge modification problems are incompressible as $\overline{H}$ is 3-connected for most such H. In fact for trees H, we know that H-free edge modification problems are incompressible for all but a small number of trees [7]. In this regards, $H = K_{1,3}$ (the claw graph) is a very challenging case.

Problem 20. *Determine whether claw-free edge modification problems admit polynomial kernels.*

For general H, we pretty much know how blocks and connected components in H affect the incompressibility of H-free modification problems [7]. This leaves 2-connected H a very important case. Note that DIAMOND-FREE EDGE DELETION admits a polynomial kernel [7].

Conjecture 21. *For any fixed 2-connected H, H-FREE EDGE DELETION and EDITING are incompressible unless H is complete or the diamond graph $K_4 - e$, and H-FREE EDGE COMPLETION is incompressible unless H misses at most one edge.*

Since most hereditary families of graphs are characterized by several forbidden subgraphs, it is also meaningful and important to study the incompressibility of their corresponding edge modification problems.

Problem 22. *Let $\mathcal{F}$ be a family of graphs. What is the relation between the incompressibility of $\mathcal{F}$-free edge modification problems and that of H-free edge modification problems for every $H \in \mathcal{F}$? In particular, does the incompressibility of H_1- and H_2-free edge modification problems imply that of $\{H_1, H_2\}$-free edge modification problem?*

We hope that our work in the paper will be useful towards a dichotomy theorem on incompressibility of H-free edge modification problems, or perhaps even a dichotomy theorem for the general $\mathcal{F}$-free edge modification problems.

References

1. Alon, N., Gutin, G., Kim, E.J., Szeider, S., Yeo, A.: Solving Max-r-SAT above a tight lower bound. Algorithmica 61, 638–655 (2011)
2. Bodlaender, H.L., Cai, L., Chen, J., Fellows, M.R., Telle, J.A., Marx, D.: Open problems in parameterized and exact computation — IWPEC 2006. Utrecht University Technical Report UU-CS-2006-052 (2006)

3. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. Journal of Computer and System Sciences 75(8), 423–434 (2009)
4. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Cross-composition: a new technique for kernelization lower bounds. arXiv:1011.4224v2 (2012), `http://arxiv.org/abs/1011.4224`
5. Bodlaender, H.L., Thomassé, S., Yeo, A.: Analysis of data reduction: transformations give evidence for non-existence of polynomial kernels. Utrecht University Technical Report UU-CS-2008-030 (2008)
6. Cai, L.: Fixed-parameter tractability of graph modification problems for hereditary properties. Information Processing Letters 58(4), 157–206 (1996)
7. Cai, Y.: Polynomial Kernelisation of H-free Edge Modification Problems. MPhil Thesis, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China (2012), `http://www.uni-marburg.de/fb12/ps/team/cai-masterarbeit.pdf`
8. Fortnow, L., Santhanam, R.: Infeasibility of instance compression and succinct PCPs for NP. Journal of Computer and System Sciences 77(1), 91–106 (2011)
9. Gramm, J., Guo, J., Hüffner, F., Niedermeier, R.: Graph-modeled data clustering: fixed parameter algorithms for clique generation. Theory of Computing Systems 38(4), 373–392 (2005)
10. Guillemot, S., Paul, C., Perez, A.: On the (non-)existence of polynomial kernels for P_l-free edge modification problems. In: Raman, V., Saurabh, S. (eds.) IPEC 2010. LNCS, vol. 6478, pp. 147–157. Springer, Heidelberg (2010)
11. Kratsch, S., Wahlström, M.: Two edge modification problems without polynomial kernels. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 264–275. Springer, Heidelberg (2009)