

# Hardness of Approximation for $H$ -Free Edge Modification Problems: Towards a Dichotomy

Tatiana Belova ✉

St. Petersburg Department of Steklov Mathematical Institute of the RAS, Russia

Ivan Bliznets ✉

St. Petersburg Department of Steklov Mathematical Institute of the RAS, Russia

---

## Abstract

For a fixed graph  $H$ , the  $H$ -free Edge Deletion/Completion/Editing problem asks to delete/add/edit the minimum possible number of edges in  $G$  to get a graph that does not contain an induced subgraph isomorphic to the graph  $H$ . In this work, we investigate  $H$ -free Edge Deletion/Completion/Editing problems in terms of the hardness of their approximation. We formulate a conjecture according to which all the graphs with at least five vertices can be classified into several groups of graphs with specific structural properties depending on the hardness of approximation for the corresponding  $H$ -free Edge Deletion/Completion/Editing problems. Also, we make significant progress in proving that conjecture by showing that it is sufficient to resolve it only for a finite set of graphs.

**2012 ACM Subject Classification** Theory of computation → Fixed parameter tractability

**Keywords and phrases** Parameterized complexity, Hardness of approximation, Edge modification problems

**Digital Object Identifier** 10.4230/LIPIcs.ISAAC.2022.24

**Related Version** *Full Version*:

<https://drive.google.com/file/d/1jgaKmHye9dABKlt-6jt9haVcL4cNAPDC/view?usp=sharing>

**Funding** The research is supported by Russian Science Foundation (project 18-71-10042).

## 1 Introduction

A graph modification problem is usually formulated in the following way. There is a fixed set of allowed graph modifications, e.g., removing vertices or edges. Given a graph  $G$ , the problem asks to find the minimum number of allowed modifications such that the resulting graph satisfies a certain property. Some of the well-known graph problems can be formulated as graph modification problems. For example, in the Vertex Cover problem, the task is to remove the minimum possible number of vertices to make the resulting graph contain no edges. Usually, graph modification problems are NP-hard. In [19], Lewis and Yannakakis showed that if the property that a graph should satisfy after modifications is non-trivial (that means that it holds for an infinite set of graphs and does not hold either for an infinite set of graphs) and inherited (that means that if a graph  $G$  satisfies that property then each induced subgraph of  $G$  also satisfies that property), and the only allowed modification is removing vertices, then such problems are NP-hard. However, for edge modification problems the situation is more complicated. In the early works [21, 12] devoted to edge modification problems, it has been observed that although we can obtain results for specific classes of problems, it can be difficult to generalize them to all edge modification problems.

Graph modification problems are extensively studied from the point of view of parameterized complexity [1, 4, 5, 7, 9, 11, 13, 16, 20], because for practical reasons the number of allowed modifications is usually small compared to the size of the graph, and then it is reasonable to consider the number of modifications as a parameter. For a detailed survey of parameterized algorithms for modification problems we recommend [9].

One of the typical representatives of graph modification problems are  $H$ -free Edge Deletion,  $H$ -free Edge Completion and  $H$ -free Edge Editing for every fixed graph  $H$ . In the  $H$ -free Edge Deletion problem, given a graph  $G$ , the task is to remove the minimum



© Tatiana Belova and Ivan Bliznets;

licensed under Creative Commons License CC-BY 4.0

33rd International Symposium on Algorithms and Computation (ISAAC 2022).

Editors: Sang Won Bae and Heejin Park; Article No. 24; pp. 24:1–24:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

number of edges from  $G$  such that the resulting graph does not contain an induced subgraph isomorphic to  $H$ . Similarly, in  $H$ -free Edge Completion we need to add the smallest number of edges and in  $H$ -free Edge Editing we are allowed to use both edge deletion and edge completion.

Using a branching method, Cai [6] showed that  $H$ -free Edge Deletion (Completion, Editing) can be solved in  $c^k \cdot \text{poly}(n)$  time, where  $k$  is an upper bound for the number of allowed modifications,  $n$  is the size of graph, and  $c$  depends only on the graph  $H$ , which is fixed for each problem. Also, in general it is unlikely that  $H$ -free Edge Deletion (Completion, Editing) can be solved in time significantly less than  $c^k \cdot \text{poly}(n)$ , because Aravind et al. [1] showed that for all graphs  $H$  with at least two edges (non-edges),  $H$ -free Edge Deletion (Completion) are NP-hard and cannot be solved in  $2^{o(k)} \cdot \text{poly}(n)$  time, if the ETH holds, and for all graphs  $H$  with at least three vertices,  $H$ -free Edge Editing is NP-hard and cannot be solved in  $2^{o(k)} \cdot \text{poly}(n)$  time, if the ETH holds.

$H$ -free Edge Deletion (Completion, Editing) are also considered from the kernelization point of view, in particular, in terms of polynomial kernel existence. Kratsch and Wahlström [18] presented a graph  $H$  such that  $H$ -free Edge Deletion and  $H$ -free Edge Editing do not admit a polynomial kernel unless  $NP \subseteq coNP/poly$ . Later, Guillemot et al. [15] showed that for all graphs  $H$ , which are cycles on at least four vertices or paths on at least seven vertices,  $H$ -free Edge Deletion does not admit a polynomial kernel unless  $NP \subseteq coNP/poly$ . That result was improved in [7, 8] by Cai et al. They showed that if  $H$  is a path or a cycle, then  $H$ -free Edge Deletion (Completion, Editing) does not admit a polynomial kernel unless  $NP \subseteq coNP/poly$ , if and only if  $H$  contains at least four edges. Also, Cai et al. [7, 8] showed that if  $H$  is a 3-connected graph and  $NP \not\subseteq coNP/poly$ , then  $H$ -free Edge Deletion and  $H$ -free Edge Editing do not admit a polynomial kernel if and only if  $H$  is not a complete graph, and  $H$ -free Edge Completion does not admit a polynomial kernel if and only if  $H$  contains at least two non-edges. Marx and Sandeep [20] made an attempt to classify all graphs  $H$  on at least five vertices depending on the existence of polynomial kernels for the corresponding problems. They formulated conjectures that if  $NP \not\subseteq coNP/poly$  then, for a graph  $H$  on at least five vertices,  $H$ -free Edge Editing admits polynomial kernel if and only if  $H$  is complete or empty, and  $H$ -free Edge Deletion admits polynomial kernel if and only if  $H$  is either complete or contains at most one edge. Despite the fact that those conjectures are not yet proven, Marx and Sandeep [20] showed that to prove them it is sufficient only to show that graphs from some finite set do not admit a polynomial kernel.

$H$ -free Edge Deletion (Completion, Editing) are also studied in terms of approximation complexity. Since removing and adding edges can create new subgraphs isomorphic to  $H$ , it is unlikely to find an exact or approximate solution using greedy methods. Generally, there is a connection between the existence of polynomial kernel and  $\text{poly}(OPT)$ -approximation for a problem, because usually one can find  $\text{poly}(OPT)$ -approximation having polynomial kernel, and vice versa. Despite the fact that this connection is not formal and there are problems [14] that act differently, it is quite possible that such connection exists for the  $H$ -free Edge Deletion (Completion, Editing) problems and we can use results for kernelization complexity in the case of approximation complexity. Bliznets et al. [3] showed that for  $H$  which is a 3-connected graph with at least two non-edges, and for  $H$ , which is a cycle on at least four vertices or a path on at least five vertices,  $H$ -free Edge Deletion and  $H$ -free Edge Completion do not admit  $\text{poly}(OPT)$ -approximation unless  $P = NP$ . Those results are similar to results for kernelization complexity obtained by Cai et al. in [7, 8]. However, in contrast to the kernelization case, in case of approximation complexity, the existence of at least two non-edges in  $H$  is crucial for  $H$ -free Edge Deletion. Bliznets et al. [3] also showed

■ **Table 1** Conjecture.

	Deletion	Completion	Editing
Exact	$\overline{K_n}, \overline{K_n - e}$	$K_n, K_n - e$	—
Constant	$K_n$	$\overline{K_n}$	$\overline{K_n}, K_n$
Min Horn Deletion	$K_n - e$	$\overline{K_n - e}$	$K_n - e, \overline{K_n - e}$
no poly(OPT)	all other graphs	all other graphs	all other graphs

For each of  $H$ -free Edge Deletion/Completion/Editing problems, the table represents a partition of all graphs on at least five vertices into classes depending on the hardness of approximation for the corresponding problems.

*Deletion*, *Completion* and *Editing* columns correspond to  $H$ -free Edge Deletion,  $H$ -free Edge Completion and  $H$ -free Edge Editing problems, respectively.

*Exact* row corresponds to the existence of exact polynomial-time algorithm.

*Constant* row corresponds to the existence of constant approximation.

*Min Horn Deletion* row corresponds to Min Horn Deletion-completeness.

*no poly(OPT)* row corresponds to non-existence of  $\text{poly(OPT)}$ -approximation unless  $P = NP$ .

that  $(K_n - e)$ -free Edge Deletion problem is Min Horn Deletion-complete for every  $n \geq 5$ . Min Horn Deletion-complete problems form a separate class of problems, and for now we do not know much about them. Khanna et al. [17] showed that those problems belong to the class poly-APX, but they do not admit  $2^{\log^{1-\varepsilon} n}$ -approximation for any  $\varepsilon > 0$  unless  $P = NP$ .

**Our results.** Taking into account a significant progress in classification of  $H$ -free Edge Modification problems from the point of view of kernelization in [20] and mentioned similarities of approximation and kernelization [2], we make an attempt to classify all graphs on at least five vertices depending on approximation complexity of the corresponding  $H$ -free Edge Deletion (Completion, Editing) problems. Let us formulate a conjecture about such classification for each of the  $H$ -free Edge Deletion,  $H$ -free Edge Completion and  $H$ -free Edge Editing problems.

► **Conjecture 1.** *Let  $H$  be a graph with at least five vertices. Then, the approximation complexity of  $H$ -free Edge Deletion (Completion, Editing) can be determined according to Table 1.*

In fact, this conjecture repeats the conjectures for the case of kernelization, except that it has an additional claim about Min Horn Deletion-complete problems. Thus, on the one hand, we believe that approximation complexity and kernelization complexity for the  $H$ -free Edge Deletion (Completion, Editing) problems behave very similarly, and on the other hand, in the case of approximation complexity, it seems that we observe a slightly more complicated situation.

Unfortunately, we were unable to prove the statement of the conjecture. However, we manage to prove the following theorem.

► **Theorem 2.** *There exists a set  $\mathcal{G}$  (see Table 2) of seventeen graphs such that:*

- (i)  $\forall H \in \mathcal{G} \cup \overline{\mathcal{G}}$ ,  $H$ -free Edge Deletion (Completion) does not admit  $\text{poly(OPT)}$ -approximation  $\Leftrightarrow$  conjecture for  $H$ -free Edge Deletion (Completion) holds;
- (ii)  $\forall H \in \mathcal{G}$ ,  $H$ -free Edge Editing does not admit  $\text{poly(OPT)}$ -approximation  $\Leftrightarrow$  conjecture for  $H$ -free Edge Editing holds.

## 24:4 Hardness of Approximation for H-Free Edge Modification

We think we achieve a significant progress in classification of  $H$ -free Edge Deletion/Completion/Editing problems as well as a huge step in resolving conjecture from [3] that assumes that  $H$ -free edge modification problems behave in similar way from approximation and kernelization points of view.

As we note before, for the edge modification problems there are similarities between kernelization and approximation results. However, the proofs of approximation and kernelization results are far from being identical, one of the reasons of this stems from the fact that  $(K_n - e)$ -Free Edge Deletion is only Min Horn Deletion-complete from approximation point of view and yet it does not admit a polynomial kernel unless  $NP \subseteq coNP/poly$ . This fact could potentially lead to the existence of a significantly bigger class of graphs  $H$  for which  $H$ -Free Edge Deletion is Min Horn Deletion-complete.

It seems that from approximation point of view classification is more complicated than from kernelization point of view. For example, in [7], a full classification for  $H$  being 3-connected graph was given, while in its approximation analog paper [3] almost a full classification was given with only exception of  $H = K_n - e$ . It was shown that  $(K_n - e)$ -Free Edge Deletion is Min Horn Deletion-complete, however, it neither rules out the possibility of the existence of  $poly(OPT)$ -approximation nor shows that such an approximation exists. Similarly, in [20], for full classification of kernelization it is left to prove a lack of kernel for 9 graphs for  $H$ -Free Edge Editing problem and 19 graphs for  $H$ -Free Edge Deletion/Completion problems, while the respective numbers that we achieve for approximation classification are 17 and 33 graphs.

We note that we significantly extend results for  $H$ -free Edge Deletion (Completion) from work [3], but we also consider Editing version of the  $H$ -free Edge Modification problem. The  $H$ -free Edge Editing problem was not considered in [3].

All proofs here are omitted due to space constraints, they can be found in the full version of the paper.

■ **Table 2**  $\mathcal{G} \cup \bar{\mathcal{G}}$ .

$G_1$		$\bar{G}_1$		$G_7$		$\bar{G}_7$		$G_{13}$		$\bar{G}_{13}$	
$G_2$		$\bar{G}_2$		$G_8$		$\bar{G}_8$		$G_{14}$		$\bar{G}_{14}$	
$G_3$		$\bar{G}_3$		$G_9$		$\bar{G}_9$		$G_{15}$		$\bar{G}_{15}$	
$G_4$		$\bar{G}_4$		$G_{10}$		$\bar{G}_{10}$		$G_{16}$		$\bar{G}_{16}$	
$G_5$		$\bar{G}_5$		$G_{11}$		$\bar{G}_{11}$		$G_{17}$		$\bar{G}_{17}$	
$G_6$		$\bar{G}_6$		$G_{12}$		$\bar{G}_{12}$					

**2 Preliminaries**

In this paper, we consider only simple graphs. All used notations for graphs are standard and can be found in [10]. For a fixed graph  $G$ , let  $V_\ell$  be the set of vertices in  $G$  with the minimum degree, and let  $V_h$  be the set of vertices with the maximum degree. For a set of graphs  $\mathcal{X}$ , let  $\overline{\mathcal{X}}$  be the set of complements of the graphs in  $\mathcal{X}$ . According to [20], we define the following sets of graphs:  $\mathcal{H}, \mathcal{A}, \mathcal{D}, \mathcal{B}, \mathcal{S}, \mathcal{F}$  and their complements  $\overline{\mathcal{H}}, \overline{\mathcal{A}}, \overline{\mathcal{D}}, \overline{\mathcal{B}}, \overline{\mathcal{S}}$  and  $\overline{\mathcal{F}}$ . The graphs from sets  $\mathcal{H}, \overline{\mathcal{H}}, \mathcal{A}, \overline{\mathcal{A}}, \mathcal{D}, \overline{\mathcal{D}}, \mathcal{B}, \overline{\mathcal{B}}, \mathcal{S}$  and  $\overline{\mathcal{S}}$  are shown in Tables 3, 4, 5 and 6.  $\mathcal{F}$  is a union of sets of graphs  $\mathcal{F}_i$  which are shown in Table 7. Let  $\mathcal{W}$  be the set  $\mathcal{H} \cup \overline{\mathcal{H}} \cup \mathcal{A} \cup \overline{\mathcal{A}} \cup \mathcal{D} \cup \overline{\mathcal{D}} \cup \mathcal{B} \cup \overline{\mathcal{B}} \cup \mathcal{S} \cup \overline{\mathcal{S}} \cup \mathcal{F} \cup \overline{\mathcal{F}}$ . We notice that  $\overline{\mathcal{W}} = \mathcal{W}$ .

In the  $H$ -free Edge Deletion problem, given a graph  $G$  and an integer  $k$ , we need to determine whether one can delete at most  $k$  edges from  $G$  such that the resulting graph does not contain an induced graph isomorphic to  $H$ . Let us call a set of edges  $F$  a *solution* for an instance  $(G, k)$  of  $H$ -free Edge Deletion if  $|F| \leq k$  and  $G - F$  does not contain an induced graph isomorphic to  $H$ .  $H$ -free Edge Completion and  $H$ -free Edge Editing and solutions for them are defined in a similar way. In the optimization version of the  $H$ -free Edge Deletion (Completion, Editing) problem, given a graph  $G$ , we need to determine the minimum possible  $k$  such that there exists a solution  $F$  for an instance  $(G, k)$ . We will refer to both decision and optimization versions, it should be clear from the context which version we refer to.

In the Sandwich  $H$ -free Edge Deletion problem, given a graph  $G$  and a set  $D \subseteq E(G)$  of *undeletable* edges, we need to determine whether there exists a subset  $F \subseteq E(G) \setminus D$  such that  $G - F$  does not contain  $H$  as an induced subgraph. Let us call such  $F$  a *solution* for the instance  $(G, D)$  of Sandwich  $H$ -free Edge Deletion. Sandwich  $H$ -free Edge Completion and its solution are defined in a similar way. The size of solution  $F$  is not important for us, we are interested only in its existence.

Let  $f : \mathbb{N} \rightarrow \mathbb{N}$  be a non-decreasing function. An  $f(OPT)$ -approximation for an instance  $I$  of a minimization problem  $X$  is a solution  $F$  for  $I$  such that  $|F| \leq f(OPT(I)) \cdot OPT(I)$  where  $OPT(I)$  is the size of the optimal solution for  $I$ . An algorithm for a minimization problem  $X$  is called  $f(OPT)$ -approximation algorithm if for any instance  $I$  of  $X$  it finds an  $f(OPT)$ -approximation for  $I$ . We say that a minimization problem  $X$  admits  $f(OPT)$ -approximation if there exists an  $f(OPT)$ -approximation algorithm for  $X$  working in polynomial time. It is easy to show the following results.

► **Lemma 3.**

- (i)  $H$ -free Edge Deletion admits  $f(OPT)$ -approximation  $\Leftrightarrow \overline{H}$ -free Edge Completion admits  $f(OPT)$ -approximation.
- (ii)  $H$ -free Edge Editing admits  $f(OPT)$ -approximation  $\Leftrightarrow \overline{H}$ -free Edge Editing admits  $f(OPT)$ -approximation.

■ **Table 3**  $\mathcal{H} \cup \overline{\mathcal{H}}$ .

$H_1$		$\overline{H_1}$		$H_4$		$\overline{H_4}$		$H_7$		$\overline{H_7}$	
$H_2$		$\overline{H_2}$		$H_5$		$\overline{H_5}$		$H_8$		$\overline{H_8}$	
$H_3$		$\overline{H_3}$		$H_6$		$\overline{H_6}$		$H_9$		$\overline{H_9}$	

■ Table 4  $\mathcal{A} \cup \overline{\mathcal{A}}$ .

$A_1$		$\overline{A_1}$		$A_4$		$\overline{A_4}$		$A_7$		$\overline{A_7}$	
$A_2$		$\overline{A_2}$		$A_5$		$\overline{A_5}$		$A_8$		$\overline{A_8}$	
$A_3$		$\overline{A_3}$		$A_6$		$\overline{A_6}$		$A_9$		$\overline{A_9}$	

■ Table 5  $\mathcal{D} \cup \overline{\mathcal{D}} \cup \mathcal{B} \cup \overline{\mathcal{B}}$ .

$D_1$		$\overline{D_1}$		$B_1$		$\overline{B_1}$		$B_3$		$\overline{B_3}$	
$D_2$		$\overline{D_2}$		$B_2$		$\overline{B_2}$					

► **Lemma 4.**  $\overline{K_n}$ -free Edge Deletion,  $\overline{K_n - e}$ -free Edge Deletion,  $K_n$ -free Edge Completion and  $(K_n - e)$ -free Edge Completion admit polynomial-time algorithms.

► **Lemma 5.**  $K_n$ -free Edge Deletion,  $\overline{K_n}$ -free Edge Completion,  $K_n$ -free Edge Editing and  $\overline{K_n}$ -free Edge Editing admit  $\binom{n}{2}$ -approximation.

### 3 Min Horn Deletion-complete problems

Now let us consider graphs  $H$  such that  $H$ -free Edge Deletion (Completion, Editing) is Min Horn Deletion-complete. At first, we describe which problems are called Min Horn Deletion-complete and introduce some definitions. In [17], there was an attempt to classify all constraint satisfaction problems by their approximation complexity. For some problems, the complexity of approximation was not established, but they were divided into classes in which all problems are equivalent to each other with respect to  $A$ -reduction, which we define below. Then, for each such class, we can choose any problem as a representative. Min Horn Deletion problem was chosen as one of those representatives. In Min Horn Deletion problem, we are given a Boolean formula  $\varphi$  in CNF, such that each clause of  $\varphi$  contains either a single literal or three literals where exactly one literal is negative. The task is to find an assignment that minimizes the number of unsatisfied clauses of  $\varphi$ .

We say that there is an  $A$ -reduction from a problem  $P$  to a problem  $Q$ , or  $P$   $A$ -reduces to  $Q$ , if there exist two polynomial-time algorithms  $F$  and  $G$  and a constant  $\alpha$  such that:

1. Let  $I$  be an instance of  $P$ . Then  $F(I)$  is an instance of  $Q$ .
2. Let  $I$  be an instance of  $P$ , let  $S$  be a solution for  $F(I)$ . Then  $G(I, S)$  is a solution for  $I$ .
3. Let  $I$  be an instance of  $P$ , and  $r \geq 1$ . Then, if  $S$  is an  $r$ -approximation for  $F(I)$ , then  $G(I, S)$  is an  $(\alpha r)$ -approximation for  $I$ .

We observe that  $A$ -reductions are transitive, that is if  $P$   $A$ -reduces to  $Q$ , and  $Q$   $A$ -reduces to  $R$  then  $P$   $A$ -reduces to  $R$ .

Let  $P$  and  $Q$  be two problems. We say that  $P$  is  $Q$ -complete with respect to  $A$ -reductions, or  $Q$ -complete, if there exist  $A$ -reductions from  $P$  to  $Q$  and from  $Q$  to  $P$ . Bliznets et al. [3] showed that  $(K_n - e)$ -free Edge Deletion is Min Horn Deletion-complete.

■ Table 6  $S \cup \overline{S}$ .

$S_1$		$\overline{S}_1$		$S_{13}$		$\overline{S}_{13}$		$S_{25}$		$\overline{S}_{25}$	
$S_2$		$\overline{S}_2$		$S_{14}$		$\overline{S}_{14}$		$S_{26}$		$\overline{S}_{26}$	
$S_3$		$\overline{S}_3$		$S_{15}$		$\overline{S}_{15}$		$S_{27}$		$\overline{S}_{27}$	
$S_4$		$\overline{S}_4$		$S_{16}$		$\overline{S}_{16}$		$S_{28}$		$\overline{S}_{28}$	
$S_5$		$\overline{S}_5$		$S_{17}$		$\overline{S}_{17}$		$S_{29}$		$\overline{S}_{29}$	
$S_6$		$\overline{S}_6$		$S_{18}$		$\overline{S}_{18}$		$S_{30}$		$\overline{S}_{30}$	
$S_7$		$\overline{S}_7$		$S_{19}$		$\overline{S}_{19}$		$S_{31}$		$\overline{S}_{31}$	
$S_8$		$\overline{S}_8$		$S_{20}$		$\overline{S}_{20}$		$S_{32}$		$\overline{S}_{32}$	
$S_9$		$\overline{S}_9$		$S_{21}$		$\overline{S}_{21}$		$S_{33}$		$\overline{S}_{33}$	
$S_{10}$		$\overline{S}_{10}$		$S_{22}$		$\overline{S}_{22}$		$S_{34}$		$\overline{S}_{34}$	
$S_{11}$		$\overline{S}_{11}$		$S_{23}$		$\overline{S}_{23}$		$S_{35}$		$\overline{S}_{35}$	
$S_{12}$		$\overline{S}_{12}$		$S_{24}$		$\overline{S}_{24}$		$S_{36}$		$\overline{S}_{36}$	

► **Lemma 6** ([3]). *For any  $n \geq 5$ ,  $(K_n - e)$ -free Edge Deletion is Min Horn Deletion-complete with respect to  $A$ -reductions.*

We observe that  $H$ -free Edge Deletion and  $\overline{H}$ -free Edge Completion are equivalent, so the same result follows for  $\overline{K_n - e}$ -free Edge Completion. We extend the results for editing case and prove the following lemma.

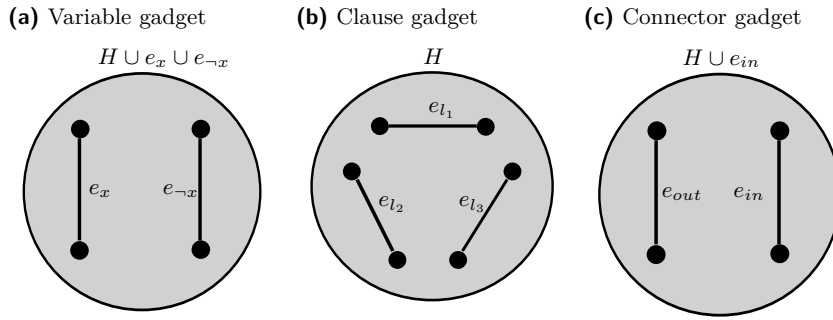
► **Lemma 7.** *For any  $n \geq 5$ ,  $(K_n - e)$ -free Edge Editing and  $\overline{K_n - e}$ -free Edge Editing are Min Horn Deletion-complete with respect to  $A$ -reductions.*

As a result, at this point we can prove statements mentioned in the first three rows of Table 1. The only remaining statements of the conjecture are the ones mentioned in the last row. We observe that if  $P = NP$  then they are trivially true, so from now on, we assume that  $P \neq NP$ .

■ Table 7  $\mathcal{F} \cup \overline{\mathcal{F}}$ .

$\mathcal{F}_1$	$K_{2,t}$	$\overline{\mathcal{F}}_1$	$K_t \cup K_2$	$t \geq 4$
$\mathcal{F}_2$	$K_{1,t}$	$\overline{\mathcal{F}}_2$	$K_t \cup K_1$	$t \geq 5$
$\mathcal{F}_3$	$K_2 \boxtimes tK_1$	$\overline{\mathcal{F}}_3$	$K_t \cup 2K_1$	$t \geq 4$
$\mathcal{F}_4$	$T_{t,1}$	$\overline{\mathcal{F}}_4$	$\overline{T}_{t,1}$	$t \geq 4$
$\mathcal{F}_5$	$\overline{(K_t - e) \cup 2K_1}$	$\overline{\mathcal{F}}_5$	$(K_t - e) \cup 2K_1$	$t \geq 4$
$\mathcal{F}_6$	$\overline{(K_t - e) \cup K_2}$	$\overline{\mathcal{F}}_6$	$(K_t - e) \cup K_2$	$t \geq 4$
$\mathcal{F}_7$	$K_{1,t} \cup K_2$	$\overline{\mathcal{F}}_7$	$\overline{K_{1,t} \cup K_2}$	$t \geq 4$
$\mathcal{F}_8$	$\overline{(K_t - e) \cup K_1}$	$\overline{\mathcal{F}}_8$	$(K_t - e) \cup K_1$	$t \geq 6$
$\mathcal{F}_9$	$J_t$	$\overline{\mathcal{F}}_9$	$\overline{J}_t$	$t \geq 3$
$\mathcal{F}_{10}$	$Q_t$	$\overline{\mathcal{F}}_{10}$	$\overline{Q}_t$	$t \geq 3$

■ Figure 1 Gadgets for Sandwich  $H$ -free Edge Deletion.



#### 4 Problems without $poly(OPT)$ -approximation

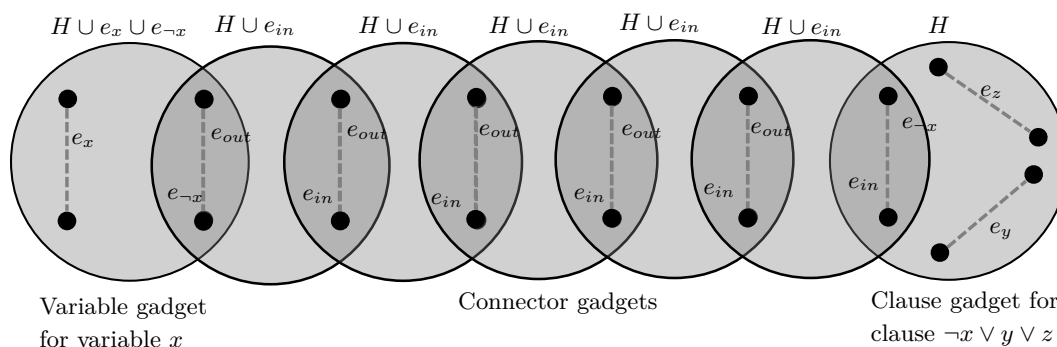
In this section, we consider several classes of graphs  $H$  such that  $H$ -free Edge Deletion (Completion, Editing) does not admit  $poly(OPT)$ -approximation. To prove it, we use a technique introduced in [3]. We also generalize that technique and use it to prove that  $H$ -free Edge Editing does not admit  $poly(OPT)$ -approximation for some graphs  $H$ .

Let us first prove  $NP$ -hardness of Sandwich  $H$ -free Edge Deletion and Sandwich  $H$ -free Edge Completion by constructing polynomial reductions from 3-SAT, which is a well-known  $NP$ -hard problem (we show those results only for  $H$  with some specific properties). In the 3-SAT problem, we are given a Boolean formula  $\varphi$  in CNF where each clause contains exactly 3 literals, and the task is to determine whether  $\varphi$  is satisfiable or not.

Let us define Variable, Clause and Connector gadgets for Sandwich  $H$ -free Edge Deletion (see Figure 1). Variable gadget for a variable  $x$  is a copy of  $H$  with two added edges  $e_x$  and  $e_{-x}$  (so  $H$  has at least two non-edges), where only these two edges are allowed to be deleted from the graph by making all other edges undeletable. Removing the edge  $e_x$  corresponds to  $x = 1$ , and removing the edge  $e_{-x}$  corresponds to  $x = 0$ . Clause gadget for a clause  $c = (\ell_1 \vee \ell_2 \vee \ell_3)$  is a copy of  $H$  where only three of its edges  $e_{l_1}$ ,  $e_{l_2}$  and  $e_{l_3}$  are allowed to be deleted by making all other edges undeletable (so  $H$  has at least three edges). Removing the edge  $e_{l_i}$  corresponds to the case when the literal  $\ell_i$  satisfies the clause  $c$ . Connector gadget is a copy of  $H$  with one added edge  $e_{in}$ , where  $e_{in}$  and some other edge  $e_{out}$  that does not share any endpoint with  $e_{in}$ , are allowed to be deleted, and all other edges are made undeletable. Using Connector gadgets, we connect Clause gadgets with Variable gadgets in such a way



■ **Figure 2** An illustration of how a Connector gadget connects a Variable gadget with a Clause gadget.



that the edges corresponding to variables and their literals are removed consistently (see Figure 2). We give the exact construction below. For Sandwich  $H$ -free Edge Completion, we define those gadgets in a similar way with slight differences, for example  $e_{in}$  becomes a non-edge as it will be on a place of some deleted edge in  $H$ .

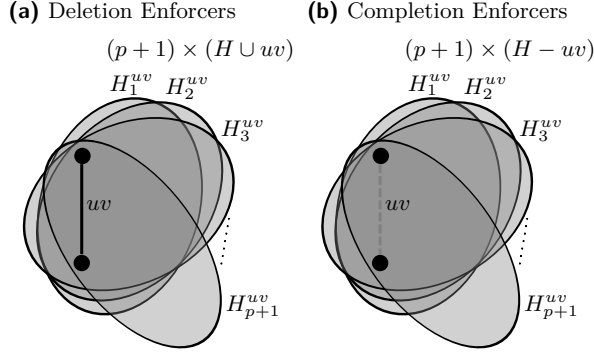
► **Construction 1.** Given an instance  $\varphi$  of 3-SAT with an assignment  $\tau$  and a set  $S = \{\text{Variable}, \text{Clause}, \text{Connector}\}$  of gadgets for Sandwich  $H$ -free Edge Deletion (Completion), we construct a graph  $G$ . For each variable we introduce its own copy of Variable gadget, and for each clause we introduce its copy of Clause gadget. For each clause  $c = (\ell_1 \vee \ell_2 \vee \ell_3)$  and each literal  $\ell \in c$ , we introduce a chain of Connector gadgets  $G_1^{\ell,c}, G_2^{\ell,c}, \dots, G_{p+2}^{\ell,c}$  where  $p = |V(H)|$ . For each  $1 \leq i \leq p+1$ , we glue  $G_i^{\ell,c}$  to  $G_{i+1}^{\ell,c}$  in such a way that the edge (non-edge)  $e_{out}$  of  $G_i^{\ell,c}$  coincide with the edge (non-edge)  $e_{in}$  of  $G_{i+1}^{\ell,c}$ . Similarly, we identify the edge (non-edge)  $e_{out}$  of  $G_{p+2}^{\ell,c}$  with the edge (non-edge)  $e_\ell$  of the Variable gadget for the variable corresponding to the literal  $\ell$ . We also identify the edge (non-edge)  $e_{in}$  of  $G_1^{\ell,c}$  with the edge (non-edge)  $e_\ell$  of the Clause gadget for the clause  $c$ . Let  $G$  be the resulting graph, and let  $D$  be the set of undeletable edges (not fillable non-edges). Let  $F \subseteq E(G)$  ( $F \subseteq \overline{E}(G)$ ) contain all edges (non-edges)  $e_\ell$  from Variable and Clause gadgets, where  $\tau(\ell) = 1$ . Also, let  $F$  contain all edges (non-edges)  $e_{in}$  and  $e_{out}$  from Connector gadgets  $G_i^{\ell,c}$ , where  $\tau(\ell) = 1$ . The result of the construction based on  $\varphi$ ,  $\tau$  and gadgets is the graph  $G \Delta F$ .

We say that  $H$  intersects a set  $\{\text{Variable}, \text{Clause}, \text{Connector}\}$  of gadgets if there exist an instance  $\varphi$  of 3-SAT and a satisfying assignment  $\tau$  for  $\varphi$ , so if  $G_\varphi^\tau$  is the result of applying Construction 1 to  $\varphi$ ,  $\tau$  and  $\{\text{Variable}, \text{Clause}, \text{Connector}\}$ , then there exists an induced subgraph of  $G_\varphi^\tau$  isomorphic to  $H$ , which is not fully contained within one gadget.

► **Lemma 8.** Let  $H$  be a graph, let  $\{\text{Variable}, \text{Clause}, \text{Connector}\}$  be a set of gadgets for Sandwich  $H$ -free Edge Deletion (Completion), and let  $H$  not intersect  $\{\text{Variable}, \text{Clause}, \text{Connector}\}$ . Then Sandwich  $H$ -free Edge Deletion (Completion) is NP-hard.

Now we introduce two more gadgets: Deletion Enforcer and Completion Enforcer. Deletion Enforcer for a graph  $H$  is a copy of  $H$  with one added edge, and Completion Enforcer for a graph  $H$  is a copy of  $H$  with one deleted edge. The added (deleted) edge we call *highlighted edge (non-edge)*. By using these two gadgets, for some graphs  $H$  we show that  $H$ -free Edge Deletion (Completion, Editing) does not admit  $\text{poly}(OPT)$ -approximation.

■ **Figure 3** Examples of how we enforce an edge to be undeletable or a non-edge to be non-fillable.



We say that  $H$  intersects Deletion (Completion) Enforcer  $X$  if there exists a graph  $G$  such that if we join  $X$  and  $G$  by identifying a highlighted edge (non-edge) of  $X$  with some edge (non-edge) of  $G$ , then the resulting graph contains an induced copy of  $H$ , which contains a vertex from  $V(X) \setminus V(G)$  and a vertex from  $V(G) \setminus V(X)$ . Gluing a large amount of Deletion (Completion) Enforcers to an instance of Sandwich  $H$ -free Edge Deletion (Completion) (see Figure 3), we can convert it into  $H$ -free Edge Deletion (Completion) in some sense. Precisely, we can prove the following lemma.

► **Lemma 9.** *Let  $H$  be a graph, let  $X$  be a Deletion (Completion) Enforcer for  $H$ , and let  $H$  not intersect  $X$ . Then (i) Sandwich  $H$ -free Edge Deletion (Completion) is NP-hard  $\Rightarrow$   $H$ -free Edge Deletion (Completion) does not admit  $\text{poly}(OPT)$ -approximation; (ii)  $H$ -free Edge Completion (Deletion) does not admit  $\text{poly}(OPT)$ -approximation  $\Rightarrow$   $H$ -free Edge Editing does not admit  $\text{poly}(OPT)$ -approximation.*

It was shown in [3] that  $H$ -free Edge Deletion and  $H$ -free Edge Completion do not admit  $\text{poly}(OPT)$ -approximation when  $H$  is a 3-connected graph with at least two non-edges or when  $H$  is a cycle with at least four vertices or a path with at least five vertices. Taking into account Lemma 3, the following lemma was shown.

► **Lemma 10** ([3]). *Let  $H$  be a 3-connected graph with at least two non-edges, or a cycle with at least four vertices or a path with at least five vertices. Then  $H$ -free Edge Deletion (Completion) and  $\bar{H}$ -free Edge Deletion (Completion) do not admit  $\text{poly}(OPT)$ -approximation.*

Using Lemma 9, we can prove a similar result for  $H$ -free Edge Editing.

► **Lemma 11.** *Let  $H$  be a 3-connected graph with at least two non-edges, or a cycle with at least four vertices, or a path with at least five vertices. Then  $H$ -free Edge Editing and  $\bar{H}$ -free Edge Editing do not admit  $\text{poly}(OPT)$ -approximation.*

Slightly adapting results from [20] for our needs, we obtain the following lemma.

► **Lemma 12.** *Let  $H$  be a regular graph which is not complete or empty. Then  $H$ -free Edge Deletion (Completion, Editing) does not admit  $\text{poly}(OPT)$ -approximation.*

## 5 General case

In this section, we show that to complete the proof of the conjecture it is sufficient to show that  $H$ -free Edge Deletion (Completion, Editing) does not admit  $\text{poly}(OPT)$ -approximation for any  $H$  from some specific set of graphs. To do this, we introduce an algorithm that, given

a graph  $H$ , constructs another graph  $H'$ , such that if  $H$ -free Edge Deletion (Completion, Editing) admits  $poly(OPT)$ -approximation then  $H'$ -free Edge Deletion (Completion, Editing) also admits  $poly(OPT)$ -approximation.

Our algorithm is similar to the algorithm given by Marx and Sandeep in [20], where they studied the existence of polynomial kernels for  $H$ -free Edge Deletion (Completion, Editing). In order to present the algorithm we introduce two sets  $\mathcal{X}$  and  $\mathcal{Y}$ .

$$\begin{aligned} \mathcal{X} &= \{C_\ell, \overline{C_\ell} \mid \ell \geq 4\} \\ &\cup \{P_\ell, \overline{P_\ell} \mid \ell \geq 5\} \\ &\cup \{H \mid H \text{ is regular but is neither complete nor empty}\} \\ &\cup \{H \mid H \text{ is 3-connected with at least two non-edges}\} \\ &\cup \{H \mid \overline{H} \text{ is 3-connected with at least two non-edges}\} \\ \mathcal{Y} &= \{K_n, \overline{K_n} \mid n \geq 5\} \\ &\cup \{K_n - e, \overline{K_n - e} \mid n \geq 5\} \\ &\cup \{H \mid H \neq C_4, \overline{H} \neq C_4, |V(H)| \leq 4\} \end{aligned}$$

Note that to finish the proof of the conjecture it is sufficient to prove that for graphs  $H \notin \mathcal{Y}$   $H$ -free Edge Deletion (Completion, Editing) does not admit  $poly(OPT)$ -approximation. Let us show that it is sufficient to prove that only for some subset of those graphs. To do this, we consider the following algorithm.

Churn( $H$ ):

- Step 1: if  $H$  is regular, then return  $H$ .
- Step 2: if  $H - V_\ell \notin \mathcal{Y}$ , then return Churn( $H - V_\ell$ ).
- Step 3: if  $H - V_h \notin \mathcal{Y}$ , then return Churn( $H - V_h$ ).
- Step 4: return  $H$ .

► **Lemma 13.** *If  $H$ -free Edge Deletion (Completion, Editing) admits  $poly(OPT)$ -approximation, then Churn( $H$ )-free Edge Deletion (Completion, Editing) also admits  $poly(OPT)$ -approximation.*

Thus, if there exists a graph  $H \notin \mathcal{Y}$  such that  $H$ -free Edge Deletion (Completion, Editing) admits  $poly(OPT)$ -approximation, then Churn( $H$ )-free Edge Deletion (Completion, Editing) also admits  $poly(OPT)$ -approximation, so it is sufficient to prove that there is no  $poly(OPT)$ -approximation only for graphs from the set  $\{\text{Churn}(H) \mid H \notin \mathcal{Y}\}$ . Our goal is to obtain a finite set of graphs for which the conjecture holds if and only if it holds for the set of all graphs with at least five vertices. In order to do that, we first consider the set  $\{\text{Churn}(H) \mid H \notin \mathcal{Y}\}$ , and then we gradually reduce it.

Let us call a set of graphs  $\mathcal{L}$  *good for Deletion (Completion, Editing) problems*, if it does not contain graphs from  $\mathcal{Y}$  and the following property holds. If  $H$  contains at least five vertices,  $H \notin \mathcal{Y}$ , and  $H$ -free Edge Deletion (Completion, Editing) admits  $poly(OPT)$ -approximation then there exists a graph  $H' \in \mathcal{L}$  such that  $H'$ -free Edge Deletion (Completion, Editing) admits  $poly(OPT)$ -approximation and  $|V(H')| \leq |V(H)|$ . We call a set  $\mathcal{L}$  *good*, if it is good for Deletion, Completion and Editing problems simultaneously. Note that both  $\{H \mid H \notin \mathcal{Y}\}$  and  $\{\text{Churn}(H) \mid H \notin \mathcal{Y}\}$  are good sets.

## 24:12 Hardness of Approximation for H-Free Edge Modification

► **Lemma 14.** *Let  $\mathcal{L}$  be good for Deletion (Completion, Editing) problems. If for each graph  $H \in \mathcal{L}$   $H$ -free Edge Deletion (Completion, Editing) does not admit  $\text{poly}(\text{OPT})$ -approximation, then for each graph  $H \notin \mathcal{Y}$   $H$ -free Edge Deletion (Completion, Editing) does not admit  $\text{poly}(\text{OPT})$ -approximation.*

Now, we want to understand which graphs form the set  $\{\text{Churn}(H) \mid H \notin \mathcal{Y}\}$ . First of all, it can be regular graphs. Observe, that we run  $\text{Churn}(H)$  only on graphs  $H$  that are not complete or empty, and we never obtain a complete or empty graph during the algorithm, since such a graph belongs to  $\mathcal{Y}$ . If we obtain a regular graph  $H$  as an output of the algorithm, then  $H$  is not complete or empty, and then  $H \in \mathcal{X}$ . If we obtain a graph  $H$  which is not regular, then we know that  $H - V_\ell \in \mathcal{Y}$  and  $H - V_h \in \mathcal{Y}$ .

Let  $\mathcal{W}$  be a set of graphs introduced in Section 2. In [20] (Lemmas 3.7 – 3.23), Marx and Sandeep proved the following lemma.

► **Lemma 15** ([20]). *Let  $H \notin \mathcal{X} \cup \mathcal{Y}$ ,  $H - V_\ell \in \mathcal{Y} \setminus \{K_n - e \mid n \geq 5\}$  and  $H - V_h \in \mathcal{Y} \setminus \{K_n - e \mid n \geq 5\}$ . Then  $H \in \mathcal{W}$ .*

This lemma covers almost all graphs  $H \notin \mathcal{X} \cup \mathcal{Y}$  such that  $H - V_\ell \in \mathcal{Y}$  and  $H - V_h \in \mathcal{Y}$ . It remains to understand what happens to graphs  $H$  for which  $\{H - V_\ell, H - V_h\} = \{K_n - e, S\}$ , where  $n \geq 5$  and  $S \in \mathcal{Y}$ . We can show that the only case when Lemma 15 is not applicable, is when  $S = \overline{K_m - e}$ , where  $m \geq 5$ . Let us call the set of such graphs  $\mathcal{U}$ . We obtain the following lemma.

► **Lemma 16.** *Let  $H \notin \mathcal{X} \cup \mathcal{Y}$ ,  $H - V_\ell \in \mathcal{Y}$  and  $H - V_h \in \mathcal{Y}$ . Then  $H \in \mathcal{W} \cup \mathcal{U}$ .*

► **Corollary 17.** *Let  $H \notin \mathcal{Y}$ . Then  $\text{Churn}(H) \in \mathcal{X} \cup \mathcal{W} \cup \mathcal{U}$ .*

Using Lemma 13, Corollary 17 and the fact that problems corresponding to graphs from  $\mathcal{X}$  do not admit  $\text{poly}(\text{OPT})$ -approximation, we obtain the following lemma.

► **Lemma 18.**  *$\mathcal{W} \cup \mathcal{U}$  is a good set.*

We note that at this stage in [20],  $\mathcal{W}$  was shown to be a good set for kernelization lower bounds. The necessity to handle graphs from  $\mathcal{U}$  is one of the key differences between approximation and kernelization cases.

From now on, we consider only graphs from  $\mathcal{W} \cup \mathcal{U}$ . Later we remove some graphs from this set and obtain a finite good set.

► **Lemma 19.** *Let  $\mathcal{L}$  be a good set for Deletion (Completion, Editing) problems. If  $H \in \mathcal{L}$  is a graph such that  $H$ -free Edge Deletion (Completion, Editing) does not admit  $\text{poly}(\text{OPT})$ -approximation, then  $\mathcal{L} \setminus \{H\}$  is a good set for Deletion (Completion, Editing) problems.*

► **Lemma 20.** *Let  $\mathcal{L}$  be a good set for Deletion (Completion, Editing) problems. Let  $H \in \mathcal{L}$ , and let  $H'$  be such a graph that the fact that  $H$ -free Edge Deletion (Completion, Editing) admits  $\text{poly}(\text{OPT})$ -approximation implies that  $H'$ -free Edge Deletion (Completion, Editing) admits  $\text{poly}(\text{OPT})$ -approximation. If  $H' \notin \mathcal{Y}$  and  $|V(H')| < |V(H)|$  then  $\mathcal{L} \setminus \{H\}$  is good for Deletion (Completion, Editing) problems.*

Let us call a graph  $H$  *uninteresting for Deletion (Completion, Editing) problems*, if either  $H$ -free Edge Deletion (Completion, Editing) does not admit  $\text{poly}(\text{OPT})$ -approximation or there exists a graph  $H' \notin \mathcal{Y}$  such that  $|V(H')| < |V(H)|$  and the existence of  $\text{poly}(\text{OPT})$ -approximation for  $H$ -free Edge Deletion (Completion, Editing) implies the existence of  $\text{poly}(\text{OPT})$ -approximation for  $H'$ -free Edge Deletion (Completion, Editing). Let us call a graph  $H$  *uninteresting*, if it is uninteresting for Deletion, Completion and Editing problems simultaneously. Using Lemmas 19 and 20, we obtain the following corollary.

► **Corollary 21.** *Let  $\mathcal{L}$  be a good set, and let  $H$  be an uninteresting graph. Then  $\mathcal{L} \setminus \{H\}$  is a good set.*

Thus, we have shown that we can remove uninteresting graphs from a good set and still get a good set. Now let us show that all graphs from  $\mathcal{U}$  are uninteresting and can be left out of consideration. In order to do that, we introduce some definitions.

For a fixed  $n \geq 5$ , a graph  $G = (V, E)$  is called  $K_n - e$  with  $V$ -shapes, if there exists a partition  $V = V_1 \sqcup V_2$  such that  $G[V_1]$  is isomorphic to  $K_n - e$ ,  $G[V_2]$  does not contain any edges, and each vertex in  $V_2$  has degree 2 in  $G$ . Each vertex  $v \in V_2$  we call a *peak*. A subgraph that contains  $v$ , its two neighbours and two edges incident to  $v$  we call a  $V$ -shape. A set of two vertices that are neighbours of the peak in a  $V$ -shape we call *base* of the  $V$ -shape.

► **Lemma 22.** *Let  $H$  be  $K_n - e$  with  $V$ -shapes for some  $n \geq 5$ . Let there be two  $V$ -shapes with non-intersecting bases. Then, if Sandwich  $H$ -free Edge Deletion (Completion) is NP-hard, then  $H$ -free Edge Deletion (Completion) does not admit  $\text{poly}(OPT)$ -approximation.*

► **Lemma 23.** *Let  $H$  be  $K_n - e$  with  $V$ -shapes for some  $n \geq 5$ . Then, if  $H$ -free Edge Deletion does not admit  $\text{poly}(OPT)$ -approximation, then  $H$ -free Edge Editing also does not admit  $\text{poly}(OPT)$ -approximation.*

Using those results, we can now prove that all graphs from  $\mathcal{U}$  are uninteresting. For  $H \in \mathcal{U}$ ,  $\{H - V_\ell, H - V_h\} = \{K_n - e, \overline{K_m - q}\}$ , where  $n, m \geq 5$ . We consider several cases depending on location of edge  $q$  and non-edge  $e$  in  $H$ , and show that all graphs from  $\mathcal{U}$  are uninteresting. Then, using Corollary 21, we obtain the following lemma.

► **Lemma 24.**  *$\mathcal{W}$  is a good set.*

Using the technique by Marx and Sandeep from [20], we show that all graphs from  $\mathcal{F} \cup \overline{\mathcal{F}} \cup \mathcal{S} \cup \overline{\mathcal{S}}$  are uninteresting. Hence, we prove the following lemma.

► **Lemma 25.**  *$\mathcal{H} \cup \overline{\mathcal{H}} \cup \mathcal{A} \cup \overline{\mathcal{A}} \cup \mathcal{D} \cup \overline{\mathcal{D}} \cup \mathcal{B} \cup \overline{\mathcal{B}}$  is a good set.*

Additionally, we show that  $H$ -free Edge Deletion (Completion, Editing) does not admit  $\text{poly}(OPT)$ -approximation for some of the remaining graphs. Our proofs for these graphs are constructed according to the scheme introduced earlier: we show that Sandwich  $H$ -free Edge Deletion (Completion) is NP-hard, then, using some Deletion (Completion) Enforcer, we show that  $H$ -free Edge Deletion (Completion) does not admit  $\text{poly}(OPT)$ -approximation, and then, using some Completion (Deletion) Enforcer, we show that  $H$ -free Edge Editing does not admit  $\text{poly}(OPT)$ -approximation.

As a result, we show that graphs  $A_2, \overline{A_2}, A_3, \overline{A_3}, A_4 = \overline{A_4}, A_5 = \overline{A_5}, B_1, \overline{B_1}, B_3, \overline{B_3}$  are uninteresting. So, the set of graphs  $\mathcal{G} \cup \overline{\mathcal{G}}$  presented in Table 2 is a good set. Observe that, by Lemma 3, if  $\forall H \in \mathcal{G}$   $H$ -free Edge Editing does not admit  $\text{poly}(OPT)$ -approximation, then  $\forall H \in \overline{\mathcal{G}}$   $H$ -free Edge Editing also does not admit  $\text{poly}(OPT)$ -approximation. So, we have proved Theorem 2.

We note that it is left to resolve the conjecture for more graphs than in the kernelization case. In [20], Marx and Sandeep also use a two step proof. They first prove that there is no polynomial kernel for Restricted  $H$ -free Edge Deletion (Completion), which is a version of Sandwich  $H$ -free Edge Deletion (Completion) where we also want to minimize the number of deleted (added) edges. Then, using similar enforcers, they give a reduction from Restricted  $H$ -free Edge Deletion (Completion) to  $H$ -free Edge Deletion (Completion, Editing). To prove that Restricted  $H$ -free Edge Deletion (Completion) does not admit a polynomial kernel, they use a reduction from Propagational- $f$  Satisfiability on 3-regular conjunctive

formulas (every variable appears exactly three times) that does not admit a polynomial kernel. In the case of approximation, we cannot use the same trick, because Propagational- $f$  Satisfiability on 3-regular conjunctive formulas is Min Horn Deletion-complete, so it gives us only Min Horn Deletion-hardness, and we need stronger results. That is why we use a technique from [3]. But in this case, we need to show hardness for Sandwich  $H$ -free Edge Deletion (Completion) instead of Restricted  $H$ -free Edge Deletion (Completion), and it can be harder, or even impossible. For example, Sandwich  $(K_n - e)$ -free Edge Deletion admits a polynomial-time solution, while Restricted  $(K_n - e)$ -free Edge Deletion (Completion) does not admit a polynomial kernel unless  $NP \subseteq coNP/poly$  [7].

## 6 Conclusion

In this work, we consider the hardness of approximation for  $H$ -free Edge Deletion (Completion, Editing) and we formulate Conjecture 1, according to which all graphs on at least five vertices can be classified into several groups of graphs with specific structural properties depending on the hardness of approximation for  $H$ -free Edge Deletion (Completion, Editing) problems corresponding to them. We think that we make a significant progress in proving those conjectures by proving Theorem 2. To make the complete classification of  $H$ -free Edge Deletion (Completion, Editing) problems for graphs  $H$  on at least five vertices by the hardness of their approximation, it is now enough to determine the hardness of approximation for those problems only for graphs  $H \in \mathcal{G} \cup \overline{\mathcal{G}}$ , where  $\mathcal{G}$  is a set of seventeen graphs. The hardness of approximation for Min Horn Deletion-complete problems and the hardness of approximation for  $H$ -free Edge Deletion (Completion, Editing) problems for graphs  $H$  on at most four vertices are still open questions.

Also, we trace the relationship between the complexity of approximation and the complexity of kernelization for  $H$ -free Edge Deletion (Completion, Editing). We have shown that, in this case, the approaches introduced to prove that a problem does not admit a polynomial kernel can be used to prove that a problem does not admit  $poly(OPT)$ -approximation. We believe that the complexity of approximation for  $H$ -free Edge Deletion (Completion, Editing) is closely connected with the complexity of kernelization for those problems. However, in case of the approximation complexity, the situation is more complicated, because of the problems corresponding to graphs with exactly one edge or non-edge that form a separate class. So, in the paper, we make a significant step towards resolving conjecture from [3] asking whether kernelization and approximation behave in the same way for  $H$ -free Edge Modification.

---

## References

- 1 NR Aravind, RB Sandeep, and Naveen Sivadasan. Dichotomy results on the hardness of h-free edge modification problems. *SIAM Journal on Discrete Mathematics*, 31(1):542–561, 2017.
- 2 Ivan Bliznets, Marek Cygan, Paweł Komosa, Lukáš Mach, and Michał Pilipczuk. Lower bounds for the parameterized complexity of minimum fill-in and other completion problems. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1132–1151. SIAM, 2016.
- 3 Ivan Bliznets, Marek Cygan, Paweł Komosa, and Michał Pilipczuk. Hardness of approximation for h-free edge modification problems. *ACM Transactions on Computation Theory (TOCT)*, 10(2):1–32, 2018.
- 4 Ivan Bliznets, Marek Cygan, Paweł Komosa, Michał Pilipczuk, and Lukáš Mach. Lower bounds for the parameterized complexity of minimum fill-in and other completion problems. *ACM Transactions on Algorithms (TALG)*, 16(2):1–31, 2020.

- 5 Ivan Bliznets, Fedor V Fomin, Marcin Pilipczuk, and Michał Pilipczuk. Subexponential parameterized algorithm for interval completion. *ACM Transactions on Algorithms (TALG)*, 14(3):1–62, 2018.
- 6 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.
- 7 Leizhen Cai and Yufei Cai. Incompressibility of  $h$ -free edge modification problems. *Algorithmica*, 71(3):731–757, 2015.
- 8 Yufei Cai. *Polynomial kernelisation of  $H$ -free edge modification problems*. PhD thesis, Chinese University of Hong Kong, 2012.
- 9 Christophe Crespelle, Pål Grønås Drange, Fedor V Fomin, and Petr A Golovach. A survey of parameterized algorithms and the complexity of edge modification. *arXiv preprint*, 2020. [arXiv:2001.06867](https://arxiv.org/abs/2001.06867).
- 10 R. Diestel. *Graph Theory*. Electronic library of mathematics. Springer, 2006.
- 11 Pål Grønås Drange. *Parameterized graph modification algorithms*. PhD thesis, The University of Bergen, 2015.
- 12 Ehab S El-Mallah and Charles J Colbourn. The complexity of some edge deletion problems. *IEEE transactions on circuits and systems*, 35(3):354–362, 1988.
- 13 Fedor V Fomin, Petr A Golovach, and Dimitrios M Thilikos. On the parameterized complexity of graph modification to first-order logic properties. *Theory of Computing Systems*, 64(2):251–271, 2020.
- 14 Archontia C Giannopoulou, Daniel Lokshtanov, Saket Saurabh, and Ondrej Suchy. Tree deletion set has a polynomial kernel but no  $opt^{o(1)}$  approximation. *SIAM Journal on Discrete Mathematics*, 30(3):1371–1384, 2016.
- 15 Sylvain Guillemot, Frédéric Havet, Christophe Paul, and Anthony Perez. On the (non-) existence of polynomial kernels for  $p$ -free edge modification problems. *Algorithmica*, 65(4):900–926, 2013.
- 16 Haim Kaplan, Ron Shamir, and Robert E Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM Journal on Computing*, 28(5):1906–1922, 1999.
- 17 Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2001.
- 18 Stefan Kratsch and Magnus Wahlström. Two edge modification problems without polynomial kernels. In *International Workshop on Parameterized and Exact Computation*, pages 264–275. Springer, 2009.
- 19 John M Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.
- 20 Dániel Marx and RB Sandeep. Incompressibility of  $h$ -free edge modification problems: Towards a dichotomy. *Journal of Computer and System Sciences*, 125:25–58, 2022.
- 21 Mihalis Yannakakis. Edge-deletion problems. *SIAM Journal on Computing*, 10(2):297–309, 1981.