Parameterized lower bound and NP-completeness of some H-free Edge Deletion problems

N. R. Aravind[1], R. B. Sandeep[1⋆], and Naveen Sivadasan[2]

[1] Department of Computer Science & Engineering
Indian Institute of Technology Hyderabad, India
{aravind,cs12p0001}@iith.ac.in
[2] TCS Innovation Labs, Hyderabad, India
naveen@atc.tcs.com

Abstract. For a graph H, the H-free Edge Deletion problem asks whether there exist at most k edges whose deletion from the input graph G results in a graph without any induced copy of H. We prove that H-free Edge Deletion is NP-complete if H is a graph with at least two edges and every component of H is either a tree or a regular graph. Furthermore, we obtain that these NP-complete problems cannot be solved in parameterized subexponential time, i.e., in time $2^{o(k)} \cdot |G|^{O(1)}$, unless Exponential Time Hypothesis fails.

1 Introduction

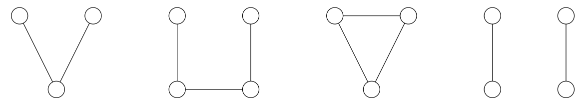
Graph modification problems ask whether we can obtain a graph G' from an input graph G by at most k number of *modifications* on G such that G' satisfies some properties. Modifications could be any kind of operations on vertices or edges. For a graph property Π, the Π Edge Deletion problem asks whether there exist at most k edges whose deletion from the input graph results in a graph with property Π. Π Edge Completion and Π Edge Editing are defined similarly, where Completion allows only adding (completing) edges and Editing allows both completion and deletion. Another graph modification problem is Π Vertex Deletion, where at most k vertex deletions are allowed. The focus of this paper is on H-free Edge Deletion. It asks whether there exist at most k edges whose removal from the input graph G results in a graph G' without any induced copy of H. The corresponding Completion problem H-free Edge Completion is equivalent to $\overline{H}$-free Edge Deletion where $\overline{H}$ is the complement graph of H. Hence the results we obtain on H-free Edge Deletion translate to that of H-free Edge Completion.

Graph modifications problems have been studied rigorously from 1970s onward. Initially, the studies were focused on proving that a modification problem is NP-complete or solvable in polynomial time. These studies resulted a good yield

⋆ supported by TCS Research Scholarship

for vertex deletion problems: Lewis and Yannakakis proved [11] that Π Vertex Deletion is NP-complete if Π is non-trivial and hereditary on induced subgraphs. In other words, Π Vertex Deletion is NP-complete if Π is defined by a finite set of forbidden induced subgraphs. Interestingly, researchers could find neither a dichotomy result similar to that of Π Vertex Deletion nor even a general hardness result for Π Edge Deletion. This scarcity of general hardness results for Π Edge Deletion is explicitly mentioned in many papers in the last four decades. For example, see [16] and [5]. It is a folklore that H-free Edge Deletion can be solved in polynomial time if H is a graph with at most one edge. Interestingly, only these H-free Edge Deletion problems are known to have polynomial time algorithms. To the best of our knowledge, only the following H-free Edge Deletion problems are known to be NP-complete: The H-free Edge Deletion problems where H is C_ℓ for any fixed $\ell \geq 3$, claw $(K_{1,3})$ [16], P_ℓ for any fixed $\ell \geq 3$ [6], $2K_2$ [4] and diamond $(K_4 - e)$ [14]. In this paper, we prove that H-free Edge Deletion is NP-complete if H has at least two edges and every component of H is either a tree or a regular graph. For every such graph H, to obtain that H-free Edge Deletion is NP-complete, we compose a series of polynomial time reductions starting from the reductions from either of the four base problems: P_3-free Edge Deletion, P_4-free Edge Deletion, K_3-free Edge Deletion and $2K_2$-free Edge Deletion. We believe that this technique can be extended to obtain a dichotomy result - H-free Edge Deletion is NP-complete if and only if H has at least two edges. The evidence for this belief is discussed in the concluding section.

Another active area of research is to give parameterized lower bounds for graph modification problems. For example, to prove that a problem cannot be solved in parameterized subexponential time, i.e., in time $2^{o(k)} \cdot |G|^{O(1)}$, under some complexity theoretic assumption. For this, the technique used is a linear parameterized reduction - a polynomial time reduction where the parameter blow up is only linear - from a problem which is already known to have no subexponential parameterized algorithm under the Exponential Time Hypothesis (ETH). ETH is a widely believed complexity theoretic assumption that 3-SAT cannot be solved in subexponential time, i.e., in time 2^{sn}, where s is a positive real number and n is the number of variables in the 3-SAT instance. Sparsification Lemma [9] states that, under ETH, there exist no algorithm to solve 3-SAT in time $2^{o(n+m)} \cdot (n+m)^{O(1)}$, where m is the number of clauses in the 3-SAT instance. Sparsification Lemma considerably helps to obtain linear parameterized reductions as it is allowed to have a parameter k such that $k = O(m+n)$ in the reduced problem. It is known that the problems mentioned in the last paragraph (H-free Edge Deletion where H is either C_ℓ for any fixed $\ell \geq 3$, claw $(K_{1,3})$ [16], P_ℓ for any fixed $\ell \geq 3$ [6], $2K_2$ [4] and diamond $(K_4 - e)$ [14]) cannot be solved in parameterized subexponential time, unless ETH fails. To the best of our knowledge, nothing more is known about the parameterized lower bound of H-free Edge Deletion problems. Since all the reductions we introduce here are linear parameterized reductions and the base problems do not admit parameterized subexponential time algorithms (unless ETH fails), we obtain that the H-free Edge Deletion cannot be solved in parameterized subexponential time if H is a graph with at least two edges and every component of H is either a tree or a regular graph, unless ETH fails.



(a) P_3 (b) P_4 (c) K_3 (d) $2K_2$

Fig. 1: The four base problems are P_3-free Edge Deletion, P_4-free Edge Deletion, K_3-free Edge Deletion and $2K_2$-free Edge Deletion.

Graph modification problems have applications in DNA physical mapping [2, 7, 8], numerical algebra [13], circuit design [6] and machine learning [1].

Outline of the Paper: Section 2 gives the notations and terminology used in the paper. It also introduces two constructions which are used for the reductions. Section 3 proves that for any tree T with at least two edges, T-free Edge Deletion is NP-complete and cannot be solved in parameterized subexponential time, unless ETH fails. Section 4 proves that for any connected regular graph R with at least two edges, R-free Edge Deletion is NP-complete and cannot be solved in parameterized subexponential time, unless ETH fails. Section 5 combines the results from Sections 3 and 4 to prove that for any graph H with at least two edges such that every component of H is either a tree or a regular graph, H-free Edge Deletion is NP-complete and cannot be solved in parameterized subexponential time, unless ETH fails. As a consequence of the equivalence between H-free Edge Deletion and $\overline{H}$-free Edge Completion, we obtain the same results for $\overline{H}$-free Edge Completion.

2 Preliminaries and Basic Tools

Graphs : We consider simple, finite and undirected graphs. The vertex set and the edge set of a graph G is denoted by $V(G)$ and $E(G)$ respectively. G is represented by the tuple $(V(G), E(G))$. A path of ℓ vertices is denoted by P_ℓ. For a vertex set $V' \subseteq V(G)$, $G[V']$ denotes the graph induced by V' in G. $G - V'$ denotes the graph obtained by deleting all the vertices in V' and the edges incident to them from G. For an edge set $E' \subseteq E(G)$, $G - E'$ denotes the graph $(V(G), E(G) \setminus E')$. The diameter of a graph G, denoted by $\text{diam}(G)$, is the number of edges in the longest induced path in G. A regular graph is a graph in which every vertex has the same degree. An r-regular graph is a graph in which every vertex has degree r. A dominating set of a graph G is a set of vertices $V' \subseteq V(G)$ such that every vertex in G is either in V' or adjacent to at least one vertex in V'. For a graph G, the disjoint union of t copies of G is denoted by tG. A component of a graph is a maximal connected subgraph. In a graph G, $\{u, v\}$ is a *non-edge* if $\{u, v\}$ is not an edge in G. We follow [15] for further notations and terminology.

Technique for Proving Parameterized Lower Bounds : Exponential Time Hypothesis (ETH) is the assumption that 3-SAT cannot be solved in time 2^{sn}, where s is a positive real number and n is the number of variables in the 3-SAT instance. Sparsification Lemma [9] states that there exists no algorithm running in time $2^{o(n+m)} \cdot (n+m)^{O(1)}$, unless ETH fails, where n and m are the number of variables and the number of clauses respectively of the 3-SAT instance. A linear parameterized reduction is a polynomial time reduction from a parameterized problem A to a parameterized problem A' such that for every instance (G, k) of A, the reduction gives an instance (G', k') of B such that $k' = O(k)$.

Proposition 1 ([3]). *If there is a linear parameterized reduction from a parameterized problem A to a parameterized problem B and if A does not admit a parameterized subexponential time algorithm, then B does not admit a parameterized subexponential time algorithm.*

We refer the book [3] for an excellent exposition on this and other aspects of parameterized algorithms and complexity.

Proposition 2. *The following problems are* NP-complete*. Furthermore, they cannot be solved in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

(i) P_3-FREE EDGE DELETION *[10]*
(ii) P_4-FREE EDGE DELETION *[4]*
(iii) C_ℓ-FREE EDGE DELETION *for any fixed $\ell \geq 3$ [16]*
(iv) $2K_2$-FREE EDGE DELETION *[4]*

In [16], Yannakakis gives a polynomial time reduction from VERTEX COVER to C_ℓ-FREE EDGE DELETION, for any fixed $\ell \geq 3$. If $\ell \neq 3$, the reduction he gives is a linear parameterized reduction. When $\ell = 3$, the reduction is not a linear parameterized reduction as it gives an instance with a parameter $k' = O(|E(G)| + k)$, where (G, k) is the VERTEX COVER instance, the input to the reduction. But, we can compose the standard 3-SAT to VERTEX COVER reduction (which is a linear parameterized reduction and gives a graph with $O(n + m)$ edges - see Theorem 3.3 in [12]) with this reduction to obtain a linear parameterized reduction from 3-SAT to K_3-FREE EDGE DELETION. For any fixed graph H, the H-FREE EDGE DELETION problem trivially belongs to NP. Hence, we may state that an H-FREE EDGE DELETION problem is NP-complete by proving that it is NP-hard.

2.1 Basic Tools

We introduce two constructions which will be used for the polynomial time reductions in the upcoming sections.

Construction 1 *Let (G', k, H, V') be an input to the construction, where G' and H are graphs, k is a positive integer and V' is a subset of vertices of H. Label the vertices of H such that every vertex get a unique label. Let the labelling be ℓ_H. For every subgraph (not necessarily induced) C with a vertex set $V(C)$ and an edge set $E(C)$ in G' such that C is isomorphic to $H[V']$, do the following:*

- *Give a labelling ℓ_C for the vertices in C such that there is an isomorphism f between C and $H[V']$ which maps every vertex v in C to a vertex u in $H[V']$ such that $\ell_C(v) = \ell_H(u)$, i.e., $f(v) = u$ if and only if $\ell_C(v) = \ell_H(u)$.*
- *Introduce $k+1$ sets of vertices $V_1, V_2 \ldots V_{k+1}$ each of size $|V(H) \setminus V'|$.*
- *For each set V_i, introduce an edge set E_i of size $|E(H) \setminus E(H[V'])|$ among $V_i \cup V(C)$ such that there is an isomorphism h between H and $(V(C) \cup V_i, E(C) \cup E_i)$ which preserves f, i.e., for every vertex $v \in V(C)$, $h(v) = f(v)$.*

This completes the construction. Let the constructed graph be G.

Let C be a copy of $H[V']$ in G'. Then, C is called a *base* in G'. Let $\{V_i\}$ and $\{E_i\}$ be the $k+1$ sets of vertices and $k+1$ sets of edges respectively which are introduced in the construction for the base C. Then, each V_i is called a *branch* of C and the vertices in V_i are called the *branch vertices* of C. C is called the *base* of V_i for $1 \leq i \leq k+1$. The vertex set of G' in G is denoted by $V_{G'}$.

Since H is a fixed graph, the construction runs in polynomial time. In the construction, for every base C in G', we introduce new vertices and edges such that there exist $k+1$ copies of H in G and C is the common intersection of every pair of them. This enforces that every solution of an instance (G, k) of H-free Edge Deletion is a solution of an instance (G', k) of H'-free Edge Deletion, where H' is $H[V']$. This is proved in the following lemma.

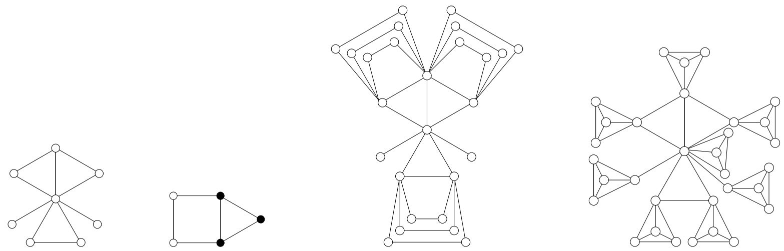
Lemma 1. *Let G be obtained by Construction 1 on the input (G', k, H, V'), where G' and H are graphs, k is a positive integer and $V' \subseteq V(H)$. Then, if (G, k) is a yes-instance of H-free Edge Deletion, then (G', k) is a yes-instance of H'-free Edge Deletion, where H' is $H[V']$.*

Proof. Let F be a solution of size at most k of (G, k). For a contradiction, assume that $G' - F$ has an induced H' with a vertex set U. Hence there is a base C in G' isomorphic to H' with the vertex set $V(C) = U$. Since there are $k+1$ copies of H in G, where each pair of copies of H has the intersection C and $|F| \leq k$, deleting F cannot kill all the copies of H associated with C. Since U induces an H' in $G' - F$, there exists a branch V_i of C such that $U \cup V_i$ induces H in $G - F$. □

Now we introduce a simple construction. This construction attaches a clique of $k+1$ vertices to each vertex in the input graph of the construction.

Construction 2 *Let (G', k) be an input to the construction, where G' is a graph and k is a positive integer. For every vertex v_i in G', introduce a set of $k+1$ vertices V_i and make every pair of vertices in $V_i \cup \{v_i\}$ adjacent. This completes the construction. Let the resultant graph be G.*

Here, we call all the newly introduced vertices as *branch vertices*.



(a) G'

(b) H. The vertices in V' are blackened.

(c) Output of Construction 1 with an input $(G', k = 2, H, V')$.

(d) Output of Construction 2 with an input $(G', k = 2)$.

Fig. 2: Examples showing Construction 1 and Construction 2.

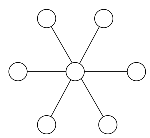
3 *T*-free Edge Deletion

Let T be any tree with at least two edges. We use induction on the diameter of T to prove that T-free Edge Deletion is NP-complete. The base cases are when $\text{diam}(T) = 2$ or 3. To prove the base cases, we use polynomial time reductions from P_3-free Edge Deletion and P_4-free Edge Deletion. For any T with $\text{diam}(T) > 3$, we give polynomial time reduction from T'-free Edge Deletion to T-free Edge Deletion, where T' is a subtree of T such that $\text{diam}(T') = \text{diam}(T) - 2$. To prove each of the base cases, we apply induction on the number of leaf vertices. All our reductions are linear parameterized reductions and hence from the non-existence of parameterized subexponential algorithms for P_3-free Edge Deletion and P_4-free Edge Deletion , we obtain that there exists no parameterized subexponential time algorithm for T-free Edge Deletion, unless ETH fails.

3.1 Base Cases

As mentioned above, the base cases are when $\text{diam}(T) = 2$ or 3. By $\ell(T)$, we denote the number of leaf vertices of T. We call the vertices in T with degree one as *leaf vertices* and the vertices with degree more than one as *internal vertices*. If $\text{diam}(T) = 2$ and $\ell(T) = \ell \geq 2$, we denote T by S_ℓ, the star graph on $\ell + 1$ vertices. We note that S_ℓ with $\ell \geq 2$ has exactly one internal vertex and ℓ leaf vertices.

For every pair of non-negative integers ℓ_1 and ℓ_2 such that $\ell_1 + \ell_2 \geq 1$, we define a tree denoted by S_{ℓ_1,ℓ_2} as follows: the vertex set V of S_{ℓ_1,ℓ_2} has $\ell_1 + \ell_2 + 2$ vertices with two designated adjacent vertices r_1 and r_2 such that r_1 is adjacent to ℓ_1 number of leaf vertices in $V \setminus \{r_2\}$ and r_2 is adjacent to ℓ_2 number of leaf vertices in $V \setminus \{r_1\}$. We call such a tree as a *twin-star* graph. We note that $S_{\ell_1,0}$ is the star graph S_{ℓ_1+1}. Both S_{ℓ_1,ℓ_2} and S_{ℓ_2,ℓ_1} are isomorphic.


(a) S_6

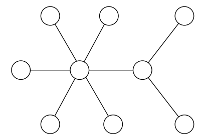

(b) $S_{5,2}$

Fig. 3: A star graph and a twin-star graph

Lemma 2. *Let $\ell > 2$. Then, $S_{\ell-1}$-FREE EDGE DELETION is polynomial time reducible to S_ℓ-FREE EDGE DELETION.*

Proof. Let (G', k) be an instance of $S_{\ell-1}$-FREE EDGE DELETION. Apply Construction 2 on (G', k) to obtain G. We claim that (G', k) is a yes-instance of $S_{\ell-1}$-FREE EDGE DELETION if and only if (G, k) is a yes-instance of S_ℓ-FREE EDGE DELETION.

Let (G', k) be a yes-instance of $S_{\ell-1}$-FREE EDGE DELETION. Let F' be a solution of size at most k of (G', k). For a contradiction, assume that $G - F'$ has an induced S_ℓ with a vertex set U. Let r be the internal vertex of the S_ℓ induced by U in $G - F'$. Now there are two cases and in both the cases we obtain contradictions.

- r is a branch vertex: Since the neighborhood of any branch vertex in $G - F'$ is a clique, r cannot be the internal vertex, which is a contradiction.
- r is a vertex in $V_{G'}$: Since the branch vertices in the neighborhood of r in $G - F'$ induce a clique, at most one branch neighbor u of r is present in U (as a leaf vertex). Hence, the remaining leaf vertices of the S_ℓ induced by U in $G - F'$ belong to $V_{G'}$. This implies that $U \setminus \{u\}$ induces $S_{\ell-1}$ in $G' - F'$, which is a contradiction.

Conversely, let (G, k) be a yes-instance of S_ℓ-FREE EDGE DELETION. Let F be a solution of size at most k of (G, k). For a contradiction, assume that $G' - F$ has an induced $S_{\ell-1}$ with a vertex set U. Let r be the internal vertex of $S_{\ell-1}$ induced by U in $G' - F$. Since $|F| \leq k$ and $k + 1$ branch vertices are adjacent to r in G, there is at least one branch vertex u adjacent to r in $G - F$. Hence, $U \cup \{u\}$ induces an S_ℓ in $G - F$, which is a contradiction. □

Theorem 1. *For any $\ell \geq 2$, S_ℓ-FREE EDGE DELETION is* NP-complete. *Furthermore, S_ℓ-FREE EDGE DELETION is not solvable in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

Proof. The proof is by induction on ℓ. When $\ell = 2$, S_ℓ is the graph P_3. Hence, Proposition 2(i) proves this case. Assume that the statements are true for $S_{\ell-1}$-FREE EDGE DELETION, if $\ell - 1 \geq 2$. Now the statements follow from Lemma 2 and the observation that the reduction used in the proof of Lemma 2 is a linear parameterized reduction. □

We apply a similar technique to prove the NP-completeness and parameterized lower bound for T-FREE EDGE DELETION when $\text{diam}(T) = 3$. As explained before, we denote these graphs by S_{ℓ_1,ℓ_2}, the twin-star graph having $\ell_1 \geq 1$ leaf vertices adjacent to an internal vertex r_1 and $\ell_2 \geq 1$ leaf vertices adjacent to another internal vertex r_2.

Lemma 3. *For integers ℓ_1 and ℓ_2 such that $\ell_1, \ell_2 \geq 1$ and $\ell_1 + \ell_2 \geq 3$, S_{ℓ_1-1,ℓ_2-1}-FREE EDGE DELETION is polynomial time reducible to S_{ℓ_1,ℓ_2}-FREE EDGE DELETION.*

Proof. Let (G', k) be an instance of S_{ℓ_1-1,ℓ_2-1}-FREE EDGE DELETION. Apply Construction 2 on (G', k) to obtain G. We claim that (G', k) is a yes-instance of S_{ℓ_1-1,ℓ_2-1}-FREE EDGE DELETION if and only if (G, k) is a yes-instance of S_{ℓ_1,ℓ_2}-FREE EDGE DELETION.

Let (G', k) be a yes-instance of S_{ℓ_1-1,ℓ_2-1}-FREE EDGE DELETION. Let F' be a solution of size at most k of (G', k). For a contradiction, assume that $G - F'$ has an induced copy of S_{ℓ_1,ℓ_2} with a vertex set U. Let r_1 and r_2 be the two internal vertices of the S_{ℓ_1,ℓ_2} induced by U in $G - F'$. Now, there are the following cases and in each case, we obtain a contradiction.

- Either r_1 or r_2 is a branch vertex: This is not possible as the neighborhood of every branch vertex induces a clique in $G - F'$.
- Both r_1 and r_2 are in $V_{G'}$: Since the branch vertices adjacent to r_1 forms a clique in $G - F'$, at most one branch vertex u_1 can be a leaf vertex adjacent to r_1 in the S_{ℓ_1,ℓ_2} induced by U in $G - F'$. Similarly, at most one branch vertex u_2 can be a leaf vertex adjacent to r_2 in the S_{ℓ_1,ℓ_2} induced by U in $G - F'$. The remaining vertices of U belong to $V_{G'}$. Hence $U \setminus \{u_1, u_2\}$ induces S_{ℓ_1-1,ℓ_2-1} in $G' - F'$, which is a contradiction.

Conversely, let (G, k) be a yes-instance of S_{ℓ_1,ℓ_2}-FREE EDGE DELETION. Let F be a solution of size at most k of (G, k). For a contradiction, assume that $G' - F$ has an induced S_{ℓ_1-1,ℓ_2-1} with a vertex set U. Since $\ell_1 + \ell_2 \geq 3$, there exists at least one internal vertex, say r_1, in the S_{ℓ_1-1,ℓ_2-1} induced by U in $G' - F$. If there is no other internal vertex r_2 in the S_{ℓ_1,ℓ_2-1}, then let r_2 be any leaf vertex of the S_{ℓ_1-1,ℓ_2-1}. Let V_1 and V_2 be the set of vertices introduced in the construction such that every vertex in V_1 is adjacent to r_1 and every vertex in V_2 is adjacent to r_2. Since $|F| \leq k$ there exist a vertex $v_1 \in V_1$ adjacent to r_1 and a vertex $v_2 \in V_2$ adjacent to r_2 in $G - F$. Hence $U \cup \{v_1, v_2\}$ induces an S_{ℓ_1,ℓ_2} in $G - F$, which is a contradiction. □

Theorem 2. *For integers ℓ_1 and ℓ_2 such that $\ell_1, \ell_2 \geq 0$ and $\ell_1 + \ell_2 \geq 1$, S_{ℓ_1,ℓ_2}-FREE EDGE DELETION is NP-complete and S_{ℓ_1,ℓ_2}-FREE EDGE DELETION is not solvable in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

Proof. The proof is by induction on $\ell_1 + \ell_2$. The base cases are:

- $\ell_1 = 0$ ($\ell_2 = 0$): This is the case when the tree is S_{ℓ_2+1} (S_{ℓ_1+1}), the case handled by Theorem 1.

- $\ell_1 = \ell_2 = 1$: Here the tree is a P_4 and hence the statements follows from Proposition 2(ii).

Assume that the statements holds true when $\ell_1 - 1, \ell_2 - 1 \geq 0$ and $(\ell_1 - 1) + (\ell_2 - 1) \geq 1$. Now, the statements follows from Lemma 3 and the observation that the reduction used in the proof of Lemma 3 is a linear parameterized reduction. □

3.2 Induction

In the last subsection, we proved the base cases of the inductive proof for the NP-completeness and parameterized lower bound of T-free Edge Deletion. The base cases were $\text{diam}(T) = 2$ (star graph) and $\text{diam}(T) = 3$ (twin-star graph). Before concluding the proof, we give a lemma which is much powerful than what we require and the implications of this lemma will be discussed in the concluding section.

Lemma 4. *Let H be any graph and d be any integer. Let V' be the set of all vertices in H with degree more than d. Let H' be $H[V']$. Then, H'-free Edge Deletion is polynomial time reducible to H-free Edge Deletion.*

Proof. Let (G', k) be an instance of H'-free Edge Deletion. Obtain G by applying Construction 1 on (G', k, H, V'). We claim that (G', k) is a yes-instance of H'-free Edge Deletion if and only if (G, k) is a yes-instance of H-free Edge Deletion.

Let (G', k) be a yes-instance of H'-free Edge Deletion. Let F be a solution of size at most k of (G', k). For a contradiction, assume that $G - F'$ has an induced H with a vertex set U. Let U' be the set of all vertices in U such that every vertex in U' has degree more than d in $(G - F')[U]$. Since every branch vertex in G has degree at most d, every vertex in U' must be in $V_{G'}$. Hence U' induces an H' in $G' - F'$, which is a contradiction.

Lemma 1 proves the converse. □

The following corollary is obtained by invoking the above lemma with $H = T$ and $d = 1$ and with the observation that the reduction used in lemma is a linear parameterized reduction.

Corollary 1. *Let T be any tree with at least two edges such that $\text{diam}(T) > 3$. Let T' be obtained from T by deleting all leaf vertices. Then, there exists a polynomial time reduction, which is a linear parameterized reduction, from T'-free Edge Deletion to T-free Edge Deletion.*

We conclude this section by the following theorem.

Theorem 3. *Let T be any tree with at least two edges. Then, T-free Edge Deletion is NP-complete. Furthermore, T-free Edge Deletion is not solvable in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

Proof. We apply induction on the diameter of T. Theorems 1 and 2 prove the statements when $\text{diam}(T) = 2$ and 3 respectively. Let the statements be true when $\text{diam}(T) = t$ for some $t \geq 3$. Assume that T has diameter $t + 1$. Deleting all leaf vertices from T gives a graph T' with diameter $t + 1 - 2 = t - 1 \geq 2$. Now, by Corollary 1, there is a linear parameterized reduction from T'-free Edge Deletion to T-free Edge Deletion. □

4 R-free Edge Deletion

In this section, we give a direct reduction either from P_3-free Edge Deletion or from K_3-free Edge Deletion to R-free Edge Deletion, for any connected r-regular graph R, where $r > 2$. The following two observations are used to prove the reduction which is given in Lemma 7.

Observation 5 *Let R be an r-regular graph for some $r > 2$. Let $V' \subseteq V(R)$ be such that $|V'| = 3$. Then, $V \setminus V'$ is a dominating set in R.*

Proof. To prove that $V \setminus V'$ is a dominating set of R, we need to prove that for every vertex $v \in V(R)$, either v is in $V \setminus V'$ or v is adjacent to a vertex in $V \setminus V'$. If $v \notin V \setminus V'$, then $v \in V'$. Since $|V'| = 3$ and v has degree $r \geq 3$, v must have at least one edge to a vertex in $V \setminus V'$. □

Observation 6 *Let G be a graph and $r > 0$ be an integer. Let $W \subseteq V(G)$ be such that every vertex in W has degree r in G and $G[W]$ is connected. Let R be any r-regular graph and G has an induced R with a vertex set W' containing at least one vertex in W. Then $W \subseteq W'$.*

Proof. Let W'' be $W \setminus W'$. For a contradiction, assume that W'' is non-empty. It is given that $W \cap W'$ is non-empty. Since $G[W]$ is connected, there exists a vertex $v \in W''$ such that v is adjacent to a vertex $u \in W \setminus W''$. Since $u \in W'$ and $G[W']$ induces an r-regular graph and u has degree r in G, we obtain that every neighbor of u must be in W'. This is a contradiction as v is a neighbor of u and is not in W'. □

Lemma 8 will prove that the assumption in the next lemma is true.

Lemma 7. *Let R be any connected r-regular graph for any $r > 2$. Assume that there exists a set of vertices $V' \subseteq V(R)$ such that $R[V']$ is a P_3 or a K_3 and $R - V'$ is connected. Let $R[V']$ be H'. Then H'-free Edge Deletion is polynomial time reducible to R-free Edge Deletion.*

Proof. Let (G', k) be an instance of H'-free Edge Deletion. We apply Construction 1 on (G', k, R, V') to obtain G. We claim that (G', k) is a yes-instance of H'-free Edge Deletion if and only if (G, k) is a yes-instance of R-free Edge Deletion.

Let F' be a solution of size at most k of (G', k). We claim that F' is a solution of (G, k). Let G'' be $G - F'$. Assume that the claim is false. Then, there is a set of vertices $U \subseteq V(G'')$ which induces R and hence there is a set of vertices $U' \subseteq U$ which induces H' in G'' such that $G''[U \setminus U']$ is a connected graph. Since $G' - F'$ is H'-free, at least one vertex $v \in U'$ must be from a branch V_j. By the construction, V_j induces a connected graph in G and hence in G''. Furthermore, every vertex in V_j has degree r in G''. Now, by Observation 6 (invoked with $G = G''$, $W = V_j$ and $W' = U$), every vertex in V_j is in U. By construction, $|V_j| = |U| - 3$. Hence, by Observation 5 (invoked with $V' = U \setminus V_j$), V_j is a dominating set in $G''[U]$. Therefore, $U = V_j \cup B_j$ where B_j is the set of base vertices of V_j in G. Since every vertex in V_j has degree r and $G''[U]$ induces an r-regular graph, every edge incident to the vertices in V_j is in $G''[U]$, i.e., $E_j \subseteq E(G''[U])$, where E_j is the edge set introduced along with V_j in Construction 1. Now by an edge counting argument, $E(G''[B_j])$ must have $|E(H')|$ number of edges. Therefore, since $|B_j| = 3$, B_j induces H' in $G' - F'$, which is a contradiction.

Lemma 1 proves the converse. □

The following lemma may be of independent interest. The assumption in Lemma 7 comes as a special case of it.

Lemma 8. *Let H be any connected graph with minimum degree d for any $d > 2$. Then there exists $V' \subseteq V(H)$ such that $|V'| = d$, $H[V']$ is connected and $H \setminus V'$ is connected.*

Proof. Let $\mathcal{H}$ be the set of all connected graphs with d number of vertices. Since the minimum degree of H is d, there exists at least one $H' \in \mathcal{H}$ as an induced subgraph of H. For a contradiction, assume that for every $V' \subseteq V(H)$ which induces any $H' \in \mathcal{H}$ in H, $H \setminus V'$ is disconnected. Consider a set of vertices $V' \subseteq V(H)$ which induces any $H' \in \mathcal{H}$ in H such that $H \setminus V'$ leaves a component with maximum number of vertices. Let the $t > 1$ components of $H \setminus V'$ be composed of set of vertices $V_1, V_2 \ldots V_t$. Without loss of generality, assume that $H[V_1]$ is a component with maximum number of vertices. Every other component has at most $d-1$ vertices. Otherwise, there will be a connected induced subgraph of d vertices in that component deleting which we get a larger component composed of $V_1 \cup V'$. Consider V_j for any j such that $2 \le j \le t$. We know that $|V_j| \le d - 1$. Hence, the degree of any vertex $v \in V_j$ is at most $d-2$ in $H[V_j]$. Since the minimum degree of H is d, there is at least 2 edges from v to V'. Let the neighbourhood of v in V' be V''. If none of the vertices in V'' is adjacent to V_1, then v and any of its $d-1$ neighbours induces a connected graph deleting which gives a larger component. If one of the vertices in V'' is adjacent to V_1, excluding that we get $d-1$ neighbours of v which along with v induces a connected subgraph deleting which gives a larger component. This is a contradiction. □

Corollary 2. *Let H be a connected graph with minimum degree 3. Then there exists an induced P_3 or K_3 with vertex set V' in H such that $H \setminus V'$ is connected.*

We conclude this section by the following theorem.

Theorem 4. *Let R be a connected regular graph with at least two edges. Then, R-*FREE EDGE DELETION *is* NP-complete*. Furthermore, R-*FREE EDGE DELETION *is not solvable in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

Proof. Let R be an r-regular graph. Since R is connected and has at least 2 edges, $r > 1$. If $r = 2$ then R is a cycle and the statements follow from Proposition 2(iii). Assume that $r \geq 3$. By Corollary 2, there exists a P_3 or K_3 with a vertex set V' in R such that $R - V'$ is connected. Now the statements follow from Lemma 7, Proposition 2(i), Proposition 2(iii) and the observation that the reduction used in the proof of Lemma 7 is a linear parameterized reduction. ☐

The complement graph of every regular graph with at least two non-edges is a regular graph with at least two edges. Thus, we obtain the following corollary.

Corollary 3. *Let R be a regular graph with at least two non-edges. Then, R-*FREE EDGE COMPLETION *is* NP-complete*. Furthermore, R-*FREE EDGE COMPLETION *is not solvable in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

5 Handling Disconnected Graphs

We have seen in Sections 3 and 4 that for any tree or connected regular graph H with at least two edges, H-FREE EDGE DELETION is NP-complete and does not admit parameterized subexponential time algorithm unless ETH fails. In this section, we extend these results to any H with at least two edges such that every component of H is either a tree or a regular graph.

Lemma 9. *Let H be a graph with $t \geq 1$ components. Let H_1 be a component of H with maximum number of vertices. Let H' be the disjoint union of all components of H isomorphic to H_1. Then, there is a linear parameterized reduction from H'-*FREE EDGE DELETION *to H-*FREE EDGE DELETION.

Proof. Let $V' \subseteq V(H)$ be the vertex set which induces H' in H. Let (G', k) be an instance of H'-FREE EDGE DELETION. We apply Construction 1 on (G', k, H, V') to obtain G. We claim that (G', k) is a yes-instance of H'-FREE EDGE DELETION if and only if (G, k) is a yes-instance of H-FREE EDGE DELETION.

Let F' be a solution of size at most k of (G', k). For a contradiction, assume that $G - F'$ has an induced H with a vertex set U. Hence there is a vertex set $U' \subseteq U$ such that U' induces H' in $G - F'$. It is straight forward to verify that a branch vertex can never be part of an induced H' in $G - F'$. Hence U' does not contain branch vertices and hence U' induces an H' in $G' - F'$, which is a contradiction. Lemma 1 proves the converse. ☐

The following lemma handles the case of disjoint union of isomorphic connected graphs.

Lemma 10. *Let H be any connected graph. For any two integers t, s such that $t \geq s \geq 1$, there is a linear parameterized reduction from sH-*FREE EDGE DELETION *to tH-*FREE EDGE DELETION.

Proof. The proof is by induction on t. The base case when $t = s$ is trivial. Assume that the statement is true for $t-1$, if $t-1 \geq s$. Now, we give a linear parameterized reduction from $(t-1)H$-free Edge Deletion to tH-free Edge Deletion.

Let (G', k) be an instance of $(t-1)H$-free Edge Deletion. Let G'' be a disjoint union of $k+1$ copies of H. Make every pair of vertices (v_i, v_j) adjacent in G'' such that v_i and v_j are part of two different copies of H in G''. Let the resultant graph be $\hat{G}$. Let G be the disjoint union of G and $\hat{G}$. We need to prove that (G', k) is a yes-instance of $(t-1)H$-free Edge Deletion if and only if (G, k) is a yes-instance of tH-free Edge Deletion.

Let F' be a solution of size at most k of (G', k). It is straight forward to verify that $\hat{G}$ is $2H$-free. Hence, if $G - F'$ has an induced tH then $G' - F'$ has an induced $(t-1)H$, which is a contradiction. Conversely, let (G, k) be a yes-instance of tH-free Edge Deletion. Let F be a solution of size at most k of (G, k). For a contradiction, assume that $G' - F$ has an induced $(t-1)H$ with a vertex set U. Since $|F| \leq k$, F cannot kill all the induced Hs in $\hat{G}$. Hence, let $U' \subseteq V(\hat{G})$ induces H in $G - F'$. Therefore, $U \cup U'$ induces tH in $G - F$, which is a contradiction. ◻

The following corollary is obtained by invoking Lemma 10 with $s = 1$.

Corollary 4. *Let H be any connected graph. For every integer $t \geq 1$, there is a linear parameterized reduction from H-free Edge Deletion to tH-free Edge Deletion.*

The lemma given below follows from Lemmas 9 Corollary 4.

Lemma 11. *Let H be a graph such that H has a component with at least two edges. Let H_1 be a component of H with maximum number of vertices. Then there is a linear parameterized reduction from H_1-free Edge Deletion to H-free Edge Deletion.*

Proof. Let H' be the disjoint union of the components of H isomorphic to H_1. By Lemma 9, there is a linear parameterized reduction from H'-free Edge Deletion to H-free Edge Deletion. Then, by Corollary 4, there is a linear parameterized reduction from H_1-free Edge Deletion to H'-free Edge Deletion. Composing these two reductions will give a linear parameterized reduction from H_1-free Edge Deletion to H-free Edge Deletion. ◻

Theorem 5. *For every $t > 1$, tK_2-free Edge Deletion is* NP-complete*. Furthermore, tK_2-free Edge Deletion is not solvable in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

Proof. Follows directly from Proposition 2(iv) and Lemma 10. ◻

We consolidate the results of this paper in the following theorem.

Theorem 6. *Let H be any graph with at least two edges such that every component of H is either a tree or a regular graph. Then H-free Edge Deletion is* NP-complete*. Furthermore, H-free Edge Deletion is not solvable in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

Proof. If H is a tK_2 for some $t > 1$, then the statements follows from Theorem 5. If H is not tK_2 and since H has at least two edges, there exists at least one component in H with at least two edges. Let H_1 be a component of H with maximum number of vertices. Then, Lemma 11 gives a linear parameterized reduction from H_1-free Edge Deletion to H-free Edge Deletion. Since H_1 is either a tree or a connected regular graph with at least two edges, the theorem follows from Theorem 3 and Theorem 4. □

Since H-free Edge Deletion is equivalent to $\overline{H}$-free Edge Completion, we obtain the following corollary.

Corollary 5. *Let $\mathcal{H}$ be the set of all graphs H with at least two edges such that every component of H is either a tree or a regular graph. Let $\overline{\mathcal{H}}$ be the set of graphs such that a graph is in $\overline{\mathcal{H}}$ if and only if its complement is in $\mathcal{H}$. Then, for every $H \in \overline{\mathcal{H}}$, H-free Edge Completion is* NP-complete. *Furthermore, H-free Edge Completion is not solvable in time $2^{o(k)} \cdot |G|^{O(1)}$, unless ETH fails.*

6 Concluding Remarks

We proved that H-free Edge Deletion is NP-complete if H is a graph with at least two edges and every component of H is either a tree or a regular graph. We also proved that, for these graphs H, H-free Edge Deletion cannot be solved in parameterized subexponential time, unless Exponential Time Hypothesis fails. The same results apply for $\overline{H}$-free Edge Completion.

Assume that we obtain a graph H' from H by deleting every vertex with degree $\delta(H)$, the minimum degree of H. Also assume that H'-free Edge Deletion is NP-complete. Then by Lemma 4, we obtain that H-free Edge Deletion is NP-complete. The reduction in Lemma 4 is not useful if H' is a graph with at most one edge, as for this H'-free Edge Deletion is polynomial time solvable. Hence we believe that if we can prove the NP-completeness of H'-free Edge Deletion where H' is a graph in which the set of vertices with degree more than $\delta(G)$ induces a graph with at most one edge, we can prove that H-free Edge Deletion is NP-complete if and only if H has at least two edges.

Bibliography

[1] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[2] Hans L Bodlaender and Babette de Fluiter. On intervalizing k-colored graphs for DNA physical mapping. *Discrete Applied Mathematics*, 71(1):55–77, 1996.

[3] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer International Publishing, 2016.

[4] P. G. Drange, F. V. Fomin, M. Pilipczuk, and Y. Villanger. Exploring subexponential parameterized complexity of completion problems. In *STACS*, 2014.

[5] P. G. Drange and M. Pilipczuk. A polynomial kernel for trivially perfect editing. In *ESA*, 2015.

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

[7] Paul W Goldberg, Martin C Golumbic, Haim Kaplan, and Ron Shamir. Four strikes against physical mapping of DNA. *Journal of Computational Biology*, 2(1):139–152, 1995.

[8] Martin Charles Golumbic, Haim Kaplan, and Ron Shamir. On the complexity of DNA physical mapping. *Advances in Applied Mathematics*, 15(3):251–261, 1994.

[9] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 653–662. IEEE, 1998.

[10] Christian Komusiewicz and Johannes Uhlmann. Cluster editing with locally bounded modifications. *Discrete Applied Mathematics*, 160(15):2259–2270, 2012.

[11] 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.

[12] R Garey Michael and David S Johnson. Computers and intractability: A guide to the theory of NP-completeness. *WH Freeman & Co., San Francisco*, 1979.

[13] Donald J Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. *Graph theory and computing*, 183:217, 1972.

[14] R. B. Sandeep and Naveen Sivadasan. Parameterized lower bound and improved kernel for Diamond-free Edge Deletion. Under review, 2015.

[15] Douglas B. West. *Introduction to graph theory*. Prentice Hall Inc., Upper Saddle River, NJ, second edition.

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