Complexity Classification of Some Edge Modification Problems *

Assaf Natanzon † Ron Shamir †‡ Roded Sharan †§

October 1999

Abstract

In an edge modification problem one has to change the edge set of a given graph as little as possible so as to satisfy a certain property. We prove the NP-hardness of a variety of edge modification problems with respect to some well-studied classes of graphs. These include perfect, chordal, chain, comparability, split and asteroidal triple free. We show that some of these problems become polynomial when the input graph has bounded degree. We also give a general constant factor approximation algorithm for deletion and editing problems on bounded degree graphs with respect to properties that can be characterized by a finite set of forbidden induced subgraphs.

Key Words: Graph Algorithms, Edge Modification Problems, Complexity, Approximation.

1 Introduction

Problem Definition: Edge modification problems call for making small changes to the edge set of an input graph in order to obtain a graph with a desired property. These include completion, deletion and editing problems. Let Π be a graph property. In the Π-*Editing* problem the input is a graph $G = (V, E)$, and the goal is to find a minimum set $F \subseteq V \times V$ such that $G' = (V, E \triangle F)$ satisfies Π, where $E \triangle F$ denotes the symmetric difference between E and F. In the Π-*Deletion* problem only edge deletions are permitted, i.e., $F \subseteq E$. The problem is equivalent to finding a maximum subgraph of G with property Π. In the Π-*Completion* problem one is only allowed to add

*Portions of this paper appeared in the Proceedings of the Twenty-Fifth International Workshop on Graph Theoretic Concepts in Computer Science [32].

†Department of Computer Science, Tel Aviv University, Tel Aviv, Israel. {natanzon,shamir,roded}@math.tau.ac.il.

‡Supported in part by the Israel Science Foundation formed by the Israel Academy of Sciences and Humanities.

§Supported by an Eshkol fellowship from the Ministry of Science, Israel.

edges, i.e., $F \cap E = \emptyset$. Equivalently, we seek a minimum supergraph of G with property Π. In this paper we analyze edge modification problems with respect to some well-studied graph properties.

Motivation: Graph modification problems are fundamental in graph theory. Already in 1979, Garey and Johnson mentioned 18 different types of vertex and edge modification problems [12, Section A1.2]. Edge modification problems have applications in several fields, including molecular biology and numerical algebra. In many application areas a graph is used to model experimental data, and then edge modifications correspond to correcting errors in the data: Adding an edge corrects a false negative error, and deleting an edge corrects a false positive error. We summarize below some of these applications. Definitions of the graph classes are given in Section 3.

Interval modification problems have important applications in physical mapping of DNA (see [5, 9, 14, 17]). Depending on the biotechnology used and the kind of experimental errors, completion, deletion and editing problem arise, both for interval graphs and for unit interval graphs.

The chordal completion problem, which is also called the *minimum fill-in problem*, arises when numerically performing a Gaussian elimination on a sparse symmetric positive-definite matrix [34]. Since the time of the computation and its storage needs depend on the sparseness of the matrix, it is desirable to find an elimination order such that a minimum number of new non-zero elements is introduced into the matrix. Rose [34] showed that this problem is equivalent to the minimum fill-in problem.

The chordal deletion problem was proposed in trying to solve the CLIQUE problem. Some heuristics for finding a large clique (see, e.g., [37]) aim to find a maximum chordal subgraph of the input graph, on which a maximum clique can be found in polynomial time.

Previous Results: Strong negative results are known for *vertex* deletion problems: Lewis and Yannakakis [26] showed that for any property which is non-trivial and hereditary, the maximum induced subgraph problem is NP-complete. Furthermore, Lund and Yannakakis [28] proved that for any such property, and for every $\epsilon > 0$, the maximum induced subgraph problem cannot be approximated with ratio $2^{\log^{1/2-\epsilon} n}$ in quasi-polynomial time, unless $\tilde{P} = \tilde{NP}$. (Throughout we use n and m to denote the number of vertices and edges, respectively, in a graph).

For edge modification problems no such general results are known, although some attempts have been made to go beyond specific graph properties [2, 3, 11]. Most of the results obtained so far concerning edge modification problems are NP-hardness ones. (For simplicity we shall often refer to the decision version of the optimization problems). Chain Completion and Chordal Completion were shown to be NP-complete by Yannakakis in [38]. As noted in [14], the NP-completeness of Interval Completion and Unit Interval Completion also follows from [38]. Interval Completion was directly shown to be NP-complete in [12, problem GT35] and [25]. Deletion problems on interval graphs and unit interval graphs were proven to be NP-complete in [14]. Cograph Completion and Cograph

Deletion were shown to be NP-complete in [11]. Threshold Completion and Threshold Deletion were shown to be NP-complete in [30]. Comparability Completion was shown to be NP-complete in [19] and Comparability Deletion was shown to be NP-complete in [39]. The NP-completeness of Bipartite Deletion and Editing follows from the NP-completeness of the equivalent MAX-CUT problem [13].

Much fewer results are known for editing problems: Chordal Editing was proven to be NP-complete in [4]. The connected bipartite interval (caterpillar) editing problem was proven to be NP-complete in [9]. Split Editing was shown to be polynomial in [21].

Several authors studied variants of the completion problem, motivated by DNA mapping, in which the input graph is pre-colored and the required supergraph also obeys the coloring (see [5] and references thereof). Other biologically motivated problems, called *sandwich* problems, seek a supergraph satisfying a given property which does not include (pre-defined) forbidden edges. Polynomial algorithms or NP-hardness results are known for many sandwich problems [15, 18, 20, 23]. Results on the parametric complexity of several completion problems were also obtained [8, 24].

Approximation algorithms exist for several problems. In [31] an $8k$ approximation algorithm is given for the minimum fill-in problem, where k denotes the size of an optimum solution. In [1] an $O(m^{1/4}\log^{3.5} n)$ approximation algorithm is given for the minimum chordal supergraph problem (where one wishes to minimize the total number of edges in the resulting graph). For the minimum interval supergraph problem an $O(\log^2 n)$ approximation algorithm was given in [33]. In [9] it was shown that the minimum number of edge editions needed in order to convert a graph into a caterpillar cannot be approximated in polynomial time to within an additive term of $O(n^{1-\epsilon})$, for $0 < \epsilon < 1$, unless P=NP.

Contribution of this paper: In this paper we study the complexity of edge modification problems on some well-studied classes of graphs. We show, among other results, that deletion problems are NP-hard for perfect, chain, chordal, split and asteroidal triple free graphs; and that editing problems are NP-hard for perfect and comparability graphs. We also show that it is NP-hard to approximate comparability modification problems to within a factor of $18/17$. The reader is referred to Figure 1 which summarizes the complexity results for the (decision version of) modification problems that we considered.

Positive complexity results are given for bounded degree input graphs: We give a simple, general constant factor approximation algorithm for the deletion and editing problems with respect to any hereditary property that can be characterized by a finite set of forbidden induced subgraphs. We also show that Chain Deletion and Editing, Split Deletion and Threshold Deletion and Editing become polynomial when the input degrees are bounded.

Organization of the paper: Section 2 contains simple basic results that show connections between the complexity of related modification problems. Section 3 contains the main hardness results. Section 4 gives the positive

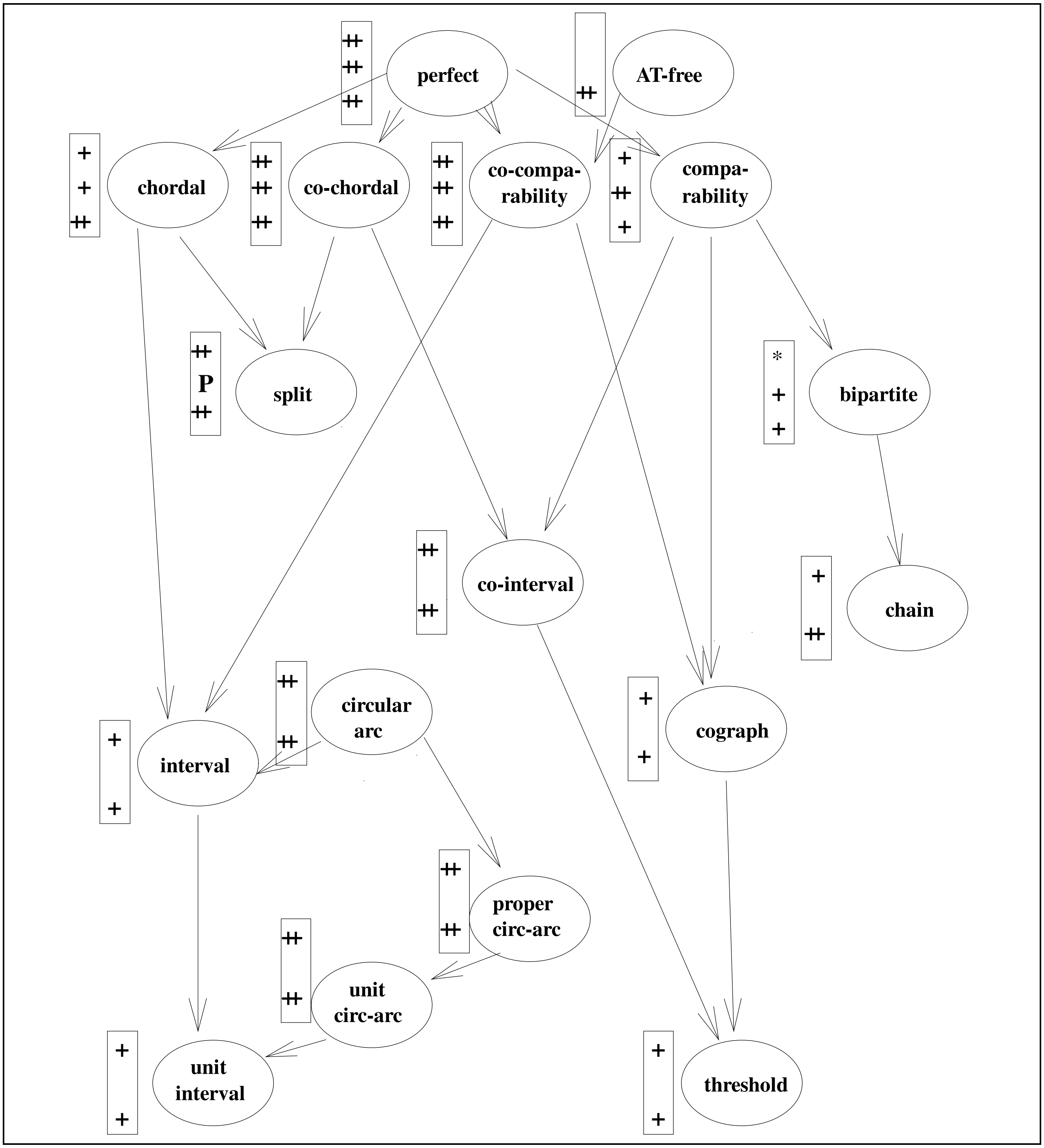

Figure 1: The complexity status of edge modification problems for some graph classes. A$\rightarrow$B indicates that class A contains class B. The box to the left of each class contains the status of the completion (top), editing (middle) and deletion (bottom) problems. $+$: NP-hard, previously known; $++$: NP-hard, new result; P: polynomial; $*$: not meaningful.

results on bounded degree graphs.

2 Basic Results

In this section we summarize some easy observations on edge modification problems, which will help us deduce complexity results from results on related graph families, and concentrate on those modification problems which are meaningful.

Definitions and Notation: All graphs in this paper are simple and contain no self-loops. Let $G = (V, E)$ be a graph. We denote its set V of vertices also by $V(G)$, and its set E of edges also by $E(G)$. We denote by $\overline{G}$ the *complement graph* of G, i.e., $\overline{G} = (V, \overline{E})$, where $\overline{E} = (V \times V) \setminus E$. (Throughout, we abuse notation for the sake of brevity, and for a set S we use $S \times S$ to denote $\{(s_1, s_2) : s_1, s_2 \in S, s_1 \neq s_2\}$.) If $G = (U, V, E)$ is bipartite then its *bipartite complement* is the bipartite graph $\overline{G} = (U, V, \overline{E})$, where $\overline{E} = (U \times V) \setminus E$. For a subset $A \subseteq V$ we denote by G_A the subgraph induced on the vertices of A. For a vertex $v \in V$ we denote by $N(v)$ the set of vertices adjacent to v in G. For any graph H, we denote by $G \cup H$ the union of the disjoint graphs G and H (with no edges connecting a vertex of G with a vertex of H). Analogously, we denote by $G + H$ the graph obtained by forming the union of G and H and connecting every vertex of G to every vertex of H. For a graph property Π the notation $G \in \Pi$ indicates that G satisfies Π. For basic definitions of graph properties and much more on the graph classes discussed here see, e.g., [7, 16].

Let Π be a graph property. If F is a set of non-edges such that $G' = (V, E \cup F) \in \Pi$ and $|F| \leq k$, then F is called a *k-completion set* with respect to Π, or a Π *k-completion set*. (For example, we shall discuss perfect k-completion set.) Π *k-deletion set* and Π *k-editing set* are similarly defined.

Basic Results: A graph property Π is called *hereditary* if when a graph G satisfies Π every induced subgraph of G satisfies Π. Π is called *hereditary on subgraphs* if when G satisfies Π, every subgraph of G satisfies Π. Π is called *ancestral* if when G satisfies Π, every supergraph of G satisfies Π.

Proposition 2.1 *If property Π is hereditary on subgraphs then Π-Deletion and Π-Editing are polynomially equivalent, and Π-Completion is not meaningful.*

Proposition 2.2 *If Π is an ancestral graph property then Π-Completion and Π-Editing are polynomially equivalent, and Π-Deletion is not meaningful.*

Proposition 2.3 *If Π and Π' are graph properties such that for every graph G and an independent set S, G satisfies Π if and only if $G \cup S$ satisfies Π', then Π-Deletion is polynomially reducible to Π'-Deletion. If in addition Π is hereditary, then Π-Completion (Π-Editing) is polynomially reducible to Π'-Completion (Π'-Editing).*

Proof: The first part of the proposition is obvious. To prove the second part we show a reduction from Π-Completion to Π'-Completion. The reduction from Π-Editing to Π'-Editing is identical. Let $< G = (V, E), k >$ be an instance of Π-Completion. We build an instance $< G' = (V', E), k >$ of Π'-Completion by adding $2k + 1$ isolated vertices to G.

We now prove validity of the reduction. If F is a Π k-completion set for G then it is also a Π' k-completion set for G', since the modified graph $(V', E \cup F)$ is a union of a graph which satisfies Π and an independent set. On the other hand, suppose that F is a Π' k-completion set for G'. Then $(V', E \cup F)$ contains an isolated vertex, and removing that vertex results in a graph satisfying Π. Since Π is hereditary, it follows that $F \cap (V \times V)$ is a Π k-completion set for G. ■

Corollary 2.4 *The following problems are NP-complete: (1) Circular-Arc Completion and Deletion; (2) Proper Circular-Arc Completion and Deletion; (3) Unit Circular-Arc Completion and Deletion.*

Proposition 2.5 *If* Π *and* Π' *are graph properties such that for every graph* G *and a clique* K*,* G *satisfies* Π *if and only if* $G + K$ *satisfies* Π'*, then* Π*-Completion is polynomially reducible to* Π'*-Completion. If in addition* Π *is hereditary, then* Π*-Deletion (*Π*-Editing) is polynomially reducible to* Π'*-Deletion (*Π'*-Editing).*

Corollary 2.6 *Permutation modification problems are polynomially reducible to the corresponding circle modification problems.*

For a graph property Π, we define the *complementary property* $\overline{\Pi}$ as follows: For every graph G, G satisfies $\overline{\Pi}$ if and only if $\overline{G}$ satisfies Π. Some well known examples are co-chordality and co-comparability.

Proposition 2.7 *For every graph property* Π*,* Π*-Deletion and* $\overline{\Pi}$*-Completion are polynomially equivalent.*

Proposition 2.8 *For every graph property* Π*,* Π*-Editing and* $\overline{\Pi}$*-Editing are polynomially equivalent.*

Corollary 2.9 *The following problems are NP-complete: (1) Co-Comparability Completion and Deletion; (2) Co-Chordal Deletion and Editing; (3) Co-Interval Completion and Deletion.*

3 NP-Hard Modification Problems

3.1 Chain Graphs

A bipartite graph $G = (P, Q, E)$ is called a *chain graph* if there is an ordering π of the vertices in P, $\pi : \{1, \ldots, |P|\} \to P$, such that $N(\pi(1)) \subseteq N(\pi(2)) \subseteq \ldots \subseteq N(\pi(|P|))$. Yannakakis introduced this class of graphs and proved that Chain Completion is NP-complete [38]. He also showed that G is a chain graph if and only if it does not contain an *independent pair of edges* (an induced $2K_2$). In this section we prove that Chain Deletion is NP-complete. This result will be the starting point to many of our subsequent reductions. Note, that in Chain Deletion (as in Chain Completion [38]) the bipartition of the input graph into P, Q is given as part of the input.

Lemma 3.1 *The bipartite complement of a chain graph is a chain graph.*

Proof: The claim follows from the observation that the chain containment order is reversed for the bipartite complement of a chain graph. Formally, let $G = (P, Q, E)$ be a chain graph, and let π be an ordering of the vertices in P such that $N(\pi(1)) \subseteq N(\pi(2)) \subseteq \ldots \subseteq N(\pi(|P|))$. Then in $\overline{G}$ we have $N(\pi(|P|)) \subseteq N(\pi(|P|-1)) \subseteq \ldots \subseteq N(\pi(1))$. ■

Corollary 3.2 *Chain Deletion is NP-complete.*

Proof: Follows from the bipartite analog of Proposition 2.7. ■

3.2 Perfect Graphs

A graph $G = (V, E)$ is called *perfect* if for every induced subgraph H of G, $\chi(H) = \omega(H)$, where $\chi(H)$ denotes the chromatic number of H, and $\omega(H)$ denotes the size of a maximum clique in H [27]. It is easy to see that a perfect graph contains no induced cycle of odd length.

Theorem 3.3 *Perfect Completion is NP-hard.*

Proof: Reduction from Chain Completion. Let $< G = (P, Q, E), k >$ be an instance of Chain Completion. We build the following instance $< P(G) = (N, E'), k >$ of Perfect Completion: Define $N = P \cup Q \cup C$, where

$$C = \{v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i} : (q_1, q_2) \in Q \times Q, 1 \leq i \leq k+1\} ,$$

and $E' = E \cup (P \times P) \cup E_1$, where

$$E_1 = \{(q_1, v^1_{q_1,q_2,i}), (v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}), (v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}), (v^3_{q_1,q_2,i}, q_2) : (q_1, q_2) \in Q \times Q, 1 \leq i \leq k+1\} .$$

In words, one side of G is transformed into a clique, and every two vertices on the other side are connected by $k+1$ disjoint 4-paths. We now prove validity of the reduction.

$\Rightarrow$ Suppose that F is a chain k-completion set for G, that is, $G' = (P, Q, E \cup F)$ is a chain graph. We claim that F is also a perfect k-completion set for $P(G)$. Let $P(G)' = (N, E' \cup F)$ and let $H = (V_H, E_H)$ be any induced subgraph of $P(G)'$. We have to show that $\omega(H) = \chi(H)$. If $E_H = \emptyset$ then H is trivially perfect, since $\chi(H) = \omega(H) = 1$. We therefore assume that $E_H \neq \emptyset$. Let $V_1 = P \cap V_H$ and let $V_2 = V_H \setminus V_1$. If $|V_1| = 0$ we can color H with two colors and $\omega(H) = \chi(H)$. Otherwise, there are two cases to examine:

1. Suppose there is a vertex in V_2 which is adjacent to all vertices in V_1. Then $w(H) \geq |V_1| + 1$. We can color H with $|V_1| + 1$ colors in the following way:
   (a) Color the vertices of V_1 with $|V_1|$ colors.
   (b) Color the vertices of Q with color number $|V_1| + 1$.
   (c) Color all vertices of type $v^2_{q_1,q_2,i}$ with color number $|V_1| + 1$.
   (d) Color all vertices of types $v^1_{q_1,q_2,i}$ and $v^3_{q_1,q_2,i}$ with color number $|V_1|$.

   Hence, $\chi(H) \leq \omega(H)$ and the claim follows (since always $\omega(H) \leq \chi(H)$).

2. If no vertex in V_2 is adjacent to all vertices in V_1, then $w(H) \geq |V_1|$ and since G' is a chain graph, there is a vertex $p \in V_1$ such that no vertex in $V_2 \cap Q$ is adjacent to p. We can color the vertices of H using $\omega(H)$ colors as follows:
   (a) Color the vertices of V_1 with $|V_1|$ colors. Let C_p denote the color of p.
   (b) Color the vertices of $V_2 \cap Q$ with C_p.
   (c) Color the vertices of type $v^2_{q_1,q_2,i}$ with C_p.
   (d) Color the vertices of types $v^1_{q_1,q_2,i}$ and $v^3_{q_1,q_2,i}$ with any existing color different from C_p.

   If $|V_1| > 1$ we used $|V_1|$ colors. If $|V_1| = 1$ we used at most two colors. In any case, $\chi(H) = \omega(H)$.

$\Leftarrow$ Suppose that F is a perfect k-completion set for $P(G)$. Let $F' = F \cap (P \times Q)$. We will show that $G' = (P, Q, E \cup F')$ is a chain graph. Suppose to the contrary that G' contains a pair of independent edges $(p_1, q_1), (p_2, q_2)$ such that $p_1, p_2 \in P$ and $q_1, q_2 \in Q$. Since $|F| \leq k$, there exists some $1 \leq i \leq k + 1$ such that the edges $(q_1, v^2_{q_1,q_2,i})$, $(q_1, v^3_{q_1,q_2,i})$, $(v^1_{q_1,q_2,i}, v^3_{q_1,q_2,i})$, $(v^1_{q_1,q_2,i}, q_2)$ and $(v^2_{q_1,q_2,i}, q_2)$ are not in F. Hence, $(N, E' \cup F)$ contains an induced cycle of odd length: If $(q_1, q_2) \in F$ then $\{q_1, v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}, q_2\}$ induce a cycle of length 5. Otherwise, $\{p_1, q_1, v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}, q_2, p_2\}$ induce a cycle of length 7. In any case we arrive at a contradiction.

■

Theorem 3.4 *Perfect Deletion is NP-hard.*

Proof: Lovász's Perfect Graph Theorem [27] states that the complement of a perfect graph is perfect. Hence, the theorem follows from Theorem 3.3 and Proposition 2.7. ■

Theorem 3.5 *Perfect Editing is NP-hard.*

Proof: Reduction from Chain Completion. Let $< G = (P, Q, E), k >$ be an instance of Chain Completion. We build the following instance $< P(G) = (N, E'), k >$ of Perfect Editing: Define $N = P \cup Q \cup C \cup D$, where

$$C = \{v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i} : (q_1, q_2) \in Q \times Q, 1 \leq i \leq k+1\} ,$$

$$D = \{w^1_{p,q,i}, w^2_{p,q,i}, w^3_{p,q,i} : (p, q) \in (P \times P) \cup E, 1 \leq i \leq k+1\} ,$$

and $E' = E \cup (P \times P) \cup E_1 \cup E_2$, where

$$E_1 = \{(q_1, v^1_{q_1,q_2,i}), (v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}), (v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}), (v^3_{q_1,q_2,i}, q_2) : (q_1, q_2) \in Q \times Q, 1 \leq i \leq k+1\} ,$$
$$E_2 = \{(p, w^1_{p,q,i}), (q, w^1_{p,q,i}), (p, w^2_{p,q,i}), (w^2_{p,q,i}, w^3_{p,q,i}), (w^3_{p,q,i}, q) : (p, q) \in E \cup (P \times P), 1 \leq i \leq k+1\} .$$

The reduction is similar to that of Theorem 3.3. The additional edges of E_2 "protect" the edges in $E \cup (P \times P)$ and prevent their removal. We now show validity of the reduction.

- $\Rightarrow$ Let F be a chain k-completion set for G. We claim that F is also a perfect k-editing set for $P(G)$. The proof is similar to the one in Theorem 3.3, and we shall use the same notation. We show below how to color the vertices of D in the two examined cases of that proof:

  1. We color all vertices of type $w^2_{p,q,i}$ with the color of q, and all vertices of type $w^3_{p,q,i}$ with the color of p. (If $p \notin V_H$ or $q \notin V_H$ we use any existing legal color.) Finally, we color all vertices of type $w^1_{p,q,i}$ with a color different than those assigned to p and q. If $|V_1| \geq 2$ then we used $|V_1| + 1 \leq \omega(H)$ colors. Otherwise, we used either 2 or 3 colors, and in any case at most $\omega(H)$ colors.
  2. We color all vertices of type $w^2_{p',q',i}$ with the color of q'. We color all vertices of type $w^3_{p',q',i}$ with the color of p'. Note, that this coloring is legal since for every $(p', q') \in E \cup (P \times P)$ the colors of p' and q' are different. This follows from the observation that p is not connected in H to any vertex in $V_2 \cap Q$. Finally, we color all vertices of type $w^1_{p',q',i}$ with a color different than those assigned to p' and q'.

- $\Leftarrow$ Let F be a perfect k-editing set for $P(G)$, and let $P(G)' = (N, E' \triangle F)$. Since $|F| \leq k$, we must have $F \cap (E \cup (P \times P)) = \emptyset$, as otherwise, $P(G)'$ would contain an induced cycle of length 5 of the form $\{p, w^1_{p,q,i}, q, w^3_{p,q,i}, w^2_{p,q,i}\}$, where $(p, q) \in E \cup (P \times P)$.

Let $F' = F \cap (P \times Q)$. By the above arguments $F' \subseteq (P \times Q) \setminus E$. We claim that F' is a k-completion set for G. Suppose to the contrary that $(P, Q, E \cup F')$ contains a pair independent edges (p_1, q_1), (p_2, q_2), where $p_1, p_2 \in P$ and $q_1, q_2 \in Q$. Since $|F| \leq k$, there exists some $1 \leq i \leq k+1$ such that the edges $(q_1, v^1_{q_1,q_2,i})$, $(v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i})$, $(v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i})$,$(v^3_{q_1,q_2,i}, q_2)$, $(q_1, v^2_{q_1,q_2,i})$,$(q_1, v^3_{q_1,q_2,i})$, $(v^1_{q_1,q_2,i}, v^3_{q_1,q_2,i})$,$(v^1_{q_1,q_2,i}, q_2)$ and $(v^2_{q_1,q_2,i}, q_2)$ are not in F. Hence, $P(G)'$ contains an induced cycle of odd length: If $(q_1, q_2) \in F$ then $\{q_1, v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}, q_2\}$ induce a cycle of length 5. Otherwise, $\{p_1, q_1, v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}, q_2, p_2\}$ induce a cycle of length 7. In any case we arrive at a contradiction.

■

3.3 Chordal Graphs

A graph is called *chordal* if it contains no induced cycle of length greater than 3 (cf. [34, 35]). We show in this section that Chordal Deletion is NP-complete.

Theorem 3.6 *Chordal Deletion is NP-complete.*

Proof: The problem is in NP since chordal graphs can be recognized in linear time [35]. We prove NP-hardness by reduction from Chain Deletion. Let $< G = (P, Q, E), k >$ be an instance of Chain Deletion. Build the following instance $< C(G) = (V', E'), k >$ of Chordal Deletion: Define $V' = P \cup Q \cup V_P \cup V_Q$, where $V_P = \{v_1, \ldots, v_k\}$ and $V_Q = \{v_{k+1}, \ldots, v_{2k}\}$. Define $E' = E \cup (P \times P) \cup (Q \times Q) \cup (P \times V_P) \cup (Q \times V_Q)$. We show that the Chordal Deletion instance has a solution if and only if the Chain Deletion instance has a solution.

- $\Rightarrow$ Suppose that F is a chain k-deletion set. We claim that F is also a chordal k-deletion set. Let $H = (V', E' \setminus F)$. Suppose to the contrary that H is not chordal, and let C be an induced cycle of length greater than 3 in H. If C contains any vertex $v \in V_P$ then the two neighbors of v on C are vertices from P, a contradiction. The same holds for V_Q. Hence, $V(C) \cap V_P = V(C) \cap V_Q = \emptyset$. Since P and Q are cliques, C must be of the form $\{p_1, p_2, q_1, q_2\}$, where $p_1, p_2 \in P$ and $q_1, q_2 \in Q$. But then (p_1, q_2) and (p_2, q_1) are independent edges in the chain graph $(P, Q, E \setminus F)$, a contradiction.

- $\Leftarrow$ Suppose that F is a chordal k-deletion set. We shall prove that $F \cap E$ is a chain k-deletion set. Let $G' = (P, Q, E \setminus F)$. If G' is not a chain graph then it contains a pair of independent edges (p_1, q_1), (p_2, q_2), where $p_1, p_2 \in P$ and $q_1, q_2 \in Q$. In $C(G)$, p_1, p_2 and also q_1, q_2 were connected by an edge and k edge-disjoint paths of length 2. Hence, each pair is still connected by a path of length at most 2 in $H = (V', E' \setminus F)$. Thus, p_1, q_1, q_2 and p_2 are on an induced cycle of length at least 4 in H, a contradiction.

■

Corollary 3.7 *Co-Chordal Completion is NP-complete.*

3.4 Split graphs

A graph G is called a *split graph* if there is a partition (K, I) of its vertex set, so that K induces a clique and I induces an independent set (cf. [16]). We prove that Split Deletion is NP-complete. Since the complement of a split graph is a split graph, this result implies that Split Completion is also NP-complete.

Theorem 3.8 *Split Deletion is NP-complete.*

Proof: Membership in NP is trivial. We prove NP-hardness by reduction from CLIQUE. Let $< G = (V, E), k >$ be an instance of CLIQUE. Build the following instance $< G' = (V', E'), k_2 = n^2(n - k + 1) - 1 >$ of Split Deletion: Define $V' = V \cup W$, where $W = \{w_1, \ldots, w_{n^2+1}\}$, and define $E' = E \cup (V \times W)$.

If G has a clique K of size at least k, then denote $K' = K \cup \{w_1\}$ and partition V' into $(K', V' \setminus K')$. The number of edges that should be deleted from G' so that it becomes a split graph with respect to this partition is at most $n^2(n - k) + \binom{n-k}{2} < n^2(n - k + 1)$.

On the other hand, suppose that G' has a k_2-deletion set, resulting in a split partition (K, I). If $|K \cap V| < k$ then at least $n^2(n - (k - 1)) > k_2$ edges in $(V \setminus K) \times (W \setminus K)$ should have been deleted from G', a contradiction.
■

Corollary 3.9 *Split Completion is NP-complete.*

3.5 AT-Free Graphs

An *asteroidal triple* is a set of three independent vertices such that there is a path between every two of them which avoids the neighborhood of the third vertex. G is called *asteroidal triple free*, or *AT-free*, if G contains no asteroidal triple [10]. Several families of graphs are asteroidal triple free, e.g., interval and co-comparability graphs. We prove in this section that AT-free Deletion is NP-complete.

Theorem 3.10 *AT-free Deletion is NP-complete.*

Proof: The problem is clearly in NP. The hardness proof is by reduction from Chain Deletion. Let $< G = (U, V, E), k >$ be an instance of Chain Deletion. Build the following instance $< A(G) = (V', E'), k >$ of AT-free Deletion: Define

$$V' = U \cup V \cup V_q \cup V_w \cup V_z \ ,$$

$$V_q = \{q_1, \ldots, q_k\}, V_w = \{w_1, \ldots, w_{k+1}\}, V_z = \{z_1, \ldots, z_{k+1}\} \ ,$$

$$E' = E \cup (U \times U) \cup (U \times V_q) \cup (U \times V_w) \cup ((V_w \cup V_z) \times (V_w \cup V_z)) \ .$$

We now prove validity of the reduction.

$\Rightarrow$ Let F be a chain k-deletion set. We claim that F is also an AT-free k-deletion set. Let $G' = (U, V, E \setminus F)$ and let $A(G)' = (V', E' \setminus F)$. Suppose to the contrary that $S = \{x, y, z\}$ is an asteroidal triple in $A(G)'$. We observe the following:

- U and $V_w \cup V_z$ remain cliques in $A(G)'$. Therefore, S contains at most one vertex from U and at most one vertex from $V_w \cup V_z$.
- For any two vertices $x, y \in V_q$, $N(x) = N(y)$. Therefore, S contains at most one vertex from V_q.
- Since G' is a chain graph, for every $x, y \in V$, $N(x) \subseteq N(y)$ or $N(y) \subseteq N(x)$. Therefore S contains at most one vertex from V.
- If S contains a vertex $u \in V_w$ then S cannot contain a vertex $v \in V_q$ since $N(v) \subseteq N(u)$.
- If $S \cap V \neq \emptyset$ then $S \cap U = \emptyset$, since every path from a vertex in V to a vertex in $V' \setminus V$ intersects the closed neighborhood of every vertex in U.
- If S contains a vertex $v \in V_q \cup V_w$ then $U \subseteq N(v)$. Therefore, $S \cap U = \emptyset$ in this case.

These observations imply that $S \cap U = \emptyset$, since otherwise S could not contain any vertex from V or from $V_q \cup V_w$, and would have therefore at most two vertices (one from U and one from V_z), a contradiction.

The only remaining possibility is that S contains a vertex from V, a vertex from V_q and a vertex from V_z, but every path from a vertex in V_z to a vertex in V intersects U, and hence, intersects the neighborhood of every vertex in V_q, a contradiction.

$\Leftarrow$ Let F be an AT-free k-deletion set. We show that $F \cap E$ is a chain k-deletion set. Let $G' = (U, V, E \setminus F)$ and let $A(G)' = (V', E' \setminus F)$. Suppose to the contrary that G' is not a chain graph. Thus, G' contains two independent edges $(u_1, v_1), (u_2, v_2)$ where $u_1, u_2 \in U$ and $v_1, v_2 \in V$. We shall prove that there is a vertex $z \in V_z$ such that $\{v_1, v_2, z\}$ is an asteroidal triple in $A(G)'$.

In $A(G)$, every vertex of U was adjacent to all $k+1$ vertices of V_w. Hence, there exist $w_1, w_2 \in V_w$, $w_1 \neq w_2$, such that $(u_1, w_1) \in E' \setminus F$ and $(u_2, w_2) \in E' \setminus F$. Similarly, there exists a vertex $z \in V_z$ such that $(w_1, z), (w_2, z) \in E' \setminus F$.

$\{v_1, v_2, z\}$ is an asteroidal triple since:

1. $\{z, w_1, u_1, v_1\}$ is a path from z to v_1 avoiding the neighborhood of v_2.
2. $\{z, w_2, u_2, v_2\}$ is a path from z to v_2 avoiding the neighborhood of v_1.
3. If $(u_1, u_2) \in E' \setminus F$ then $\{v_1, u_1, u_2, v_2\}$ is a path from v_1 to v_2 avoiding the neighborhood of z. Otherwise, there exists a vertex $q \in V_q$ such that $(u_1, q), (u_2, q) \in E' \setminus F$. Thus, $\{v_1, u_1, q, u_2, v_2\}$ is a path from v_1 to v_2 avoiding the neighborhood of z.

Hence, we arrive at a contradiction, implying that G' is a chain graph.

■

3.6 Comparability graphs

A graph is called a *comparability* graph if it has a transitive orientation of its edges, that is, an orientation F for which $(a, b), (b, c) \in F$ implies $(a, c) \in F$ (cf. [16]). We show below that Comparability Editing is NP-complete. We also prove that it is NP-hard to approximate comparability modification problems to within a factor of $18/17$.

Theorem 3.11 *Comparability Editing is NP-complete.*

Proof: Membership in NP is trivial. The hardness proof is by reduction from MAX-CUT. Given a MAX-CUT instance $< G = (V, E), k >$ we build a Comparability Editing instance $< C(G) = (N, E'), k_2 = |E| - k >$ as follows: Define $N = V \cup \{e^1_{u,v}, e^2_{u,v} : (u, v) \in E\} \cup W$, where $W = \{w^v_i : v \in V, 1 \leq i \leq 2k_2 + 1\}$. Also define $E' = E_1 \cup E_2$, where

$$\begin{aligned} E_1 &= \{(v, w^v_i) : v \in V, w^v_i \in W\} , \\ E_2 &= \{(v, e^1_{v,w}), (e^1_{v,w}, e^2_{v,w}), (e^2_{v,w}, w) : (v, w) \in E\} . \end{aligned}$$

(for each $(v, w) \in E$ the choice of which vertex to connect to $e^1_{v,w}$ is arbitrary). In words, we attach $2k_2+1$ private neighbors to each original vertex, and replace each edge by a path of length three. The validity proof follows.

- ⇒ Suppose that (V_1, V_2) is a cut of weight at least k in G, i.e., $|E \cap (V_1 \times V_2)| \geq k$. For each non-cut edge $e = (v, w) \in ((V_1 \times V_1) \cup (V_2 \times V_2)) \cap E$ we remove the edge $(e^1_{v,w}, e^2_{v,w})$ from its corresponding path in $C(G)$. In total, we remove at most k_2 edges. We now give a transitive orientation to the resulting graph, thus proving that it is a comparability graph: Orient each edge incident on $v \in V_1$ out of v, and each edge incident on $w \in V_2$ into w. For each edge $(v, w) \in (V_1 \times V_2) \cap E$, orient $(e^1_{v,w}, e^2_{v,w})$ from $e^2_{v,w}$ to $e^1_{v,w}$.

$\Leftarrow$ Suppose that F is a comparability k_2-editing set, and let $H = (N, E' \triangle F)$ be the modified comparability graph. Let R be a transitive orientation of H. For each vertex $v \in V$ its private neighbors in $N(v) \cap W$ ensure that either all edges incident on v are directed in R into v, or they are all directed out of v. (This is true since the number of private neighbors implies that at least one such neighbor a of v must remain adjacent to v only in the modified graph, and the edge (a, v) forces the orientation of all other edges.) Define a partition (V_1, V_2) of V, in which $v \in V_1$ if and only if all edges incident on v are directed into v. We shall prove that the weight of this cut is at least k. Since we modified at most $|E| - k$ edges, there are at least k paths in H of the form $\{v, e^1_{v,w}, e^2_{v,w}, w\}$, for some $(v, w) \in E$, such that no edge in F has both its endpoints in any of those paths. For each such path, its corresponding edge in G must be across the cut, as otherwise R could not have been transitive.

■

Corollary 3.12 *Co-Comparability Editing is NP-complete.*

We now prove that approximating Comparability Editing with ratio $18/17$ is NP-hard, by showing a relation between the approximability of Comparability Editing and the approximability of MAX-CUT.

Lemma 3.13 *If Comparability Editing can be approximated in polynomial time with ratio $1 + \theta$ ($\theta < 1$) then MAX-CUT can be approximated in polynomial time with ratio $\frac{1}{1-\theta}$.*

Proof: Suppose there is a $(1+\theta)$-approximation algorithm A for Comparability Editing, for some $\theta < 1$. To find an approximation for MAX-CUT on an input graph $G = (V, E)$ do the following: First, apply to G the reduction of Theorem 3.11 with $k = 0$, and obtain the graph $C(G) = (N, E')$. Second, apply algorithm A to $C(G)$ and obtain a modified comparability graph $C(G)'$. Finally, compute a cut of G as shown below.

Let $m = |E|$. Let F be a transitive orientation of $C(G)'$. For each vertex $v \in V$ its private neighbors in $N(v) \cap W$ ensure that either all edges in F are oriented into v, or all are oriented out of v. This is true since w.l.o.g. $|E(C(G)') \triangle E'| \leq m$.

Define a partition (V_1, V_2) of V, in which $v \in V_1$ if and only if all edges incident on v are directed into v. It remains to show that this cut approximates the optimum solution with ratio $\frac{1}{1-\theta}$. Let k^* be the weight of a maximum cut in G. Note, that $k^* \geq \frac{m}{2}$ since every graph has a cut which at least half of its edges cross. By the proof of Theorem 3.11 there is an editing set for $C(G)$ of size at most $m - k^*$, and therefore $|E(C(G)') \triangle E'| \leq (1 + \theta)(m - k^*)$. Thus, by the proof of Theorem 3.11 the weight of the cut (V_1, V_2) is at least

$$m - (1 + \theta)(m - k^*) = k^*(1 + \theta) - m\theta \geq k^*(1 + \theta) - 2k^*\theta = k^*(1 - \theta) \ .$$

■

Corollary 3.14 *It is NP-hard to approximate Comparability Editing to within a factor of* $18/17$.

Proof: In [22, 36] it is shown that approximating MAX-CUT to within a factor of $17/16$ is NP-hard. By Lemma 3.13 the claim follows. ■

We comment, that our reduction from MAX-CUT applies also to Comparability Completion and Comparability Deletion. Hence, it is also NP-hard to approximate the completion and deletion problems to within a factor of $18/17$.

4 Positive Results On Bounded Degree Graphs

4.1 An Approximation Algorithm for Deletion and Editing Problems

We present below a constant factor approximation algorithm for edge deletion and editing problems on bounded degree graphs. The result applies to any hereditary graph family which can be characterized by a finite set of forbidden induced subgraphs. Examples of such families include cographs, claw-free graphs and numerous others (cf. [7, Chapter 7.1]). An analogous result for vertex deletion problems was given by Yannakakis and Lund [28].

Let Π be a hereditary graph property that can be characterized by a finite set $\mathcal{F}$ of forbidden induced subgraphs. Let $G = (V, E)$ be the input graph. We assume that each forbidden subgraph contains at most t vertices and that G has maximum degree d. We first handle the case in which no forbidden subgraph contains an isolated vertex. The approximation algorithm follows.

Algorithm APPROX($G, \mathcal{F}$):
- $A \leftarrow \emptyset$
- **While** $G_{V \setminus A}$ contains an induced subgraph H isomorphic to some $F \in \mathcal{F}$, **do**:
  - $A \leftarrow A \cup V(H)$.
- Remove all edges $\{(v, w) \in E : v \in A, w \in V\}$ from G.

The algorithm is clearly polynomial since finding a forbidden induced subgraph with at most t vertices can be done in $O(n^t)$ time.

Theorem 4.1 *The algorithm approximates* Π*-Deletion and* Π*-Editing to within a factor of* td*.*

Proof: **Correctness**: After the 'while' loop is completed, $G_{V \setminus A}$ contains no forbidden induced subgraph. After the edge removal step is completed, all vertices in A become isolated. Since no forbidden induced subgraph contains an isolated vertex, at the end of the algorithm G satisfies Π.

Approximation ratio: Let F be an optimum solution of size k. For any forbidden induced subgraph H found by the algorithm, F must contain an edge with both endpoints in H. Hence, at the end of the algorithm $|A| \leq kt$, and at most ktd edges are deleted from G. ■

Now suppose that $\mathcal{F}$ contains graphs with isolated vertices, but no forbidden subgraph is an independent set. For a graph F denote by $T(F)$ the graph obtained by removing all isolated vertices from F. In case $n < 3t(d+1)$, we can solve the deletion and editing problems exactly in constant time by exhaustive search. Otherwise, we define a new set of forbidden induced subgraphs $\mathcal{F}' = \{T(F) : F \in \mathcal{F}\}$. We then apply algorithm APPROX$(G, \mathcal{F}')$. The resulting graph clearly satisfies Π, since no $F \in \mathcal{F}$ is an independent set. We analyze below the approximation ratio achieved by the algorithm.

Theorem 4.2 *The algorithm approximates* Π*-Deletion to within a factor of* td*.*

Proof: We claim that for any $F \in \mathcal{F}$ and for any d-degree bounded graph G with at least $3t(d+1)$ vertices, either G contains an induced copy of F, or G contains no induced copy of $T(F)$. Suppose to the contrary that H is an induced copy of $T(F)$ in G. Let $S \subseteq V \setminus V(H)$ denote the set of vertices which are not adjacent to any vertex of H. Since $|V(H)| \leq t-1$, $|S| \geq n-(t-1)(d+1) > 2t(d+1)$. Since G_S has degree bounded by d, it contains an independent set of size at least $\frac{2t(d+1)}{d+1} > t$. Let S' be any independent subset of S of size $|V(F)| - |V(T(F))|$. Then the vertices in $V(H) \cup S'$ induce a copy of F. This completes the proof of the claim. The approximation ratio now follows from the same arguments as in the proof of Theorem 4.1. ■

To prove the same result for Π-Editing we need the following lemma:

Lemma 4.3 *A graph with a maximum independent set of size* l *has least* $\frac{(n-l)\cdot(n-2l)}{2l}$ *edges.*

Proof: Let $G = (V, E)$ be a graph, and let S be a maximum independent set of G, $|S| \leq l$. Necessarily, each vertex in $V \setminus S$ is adjacent to some vertex in S. Therefore, there are at least $n - l$ edges between S and $V \setminus S$.

The induced subgraph $G_{V\setminus S}$ satisfies the same property. By induction, the number of edges in G is at least

$$\sum_{r=1}^{\lfloor \frac{n}{l} \rfloor}(n-rl) \geq l\binom{\lfloor \frac{n}{l} \rfloor}{2} \geq \frac{l(\frac{n}{l}-1)\cdot(\frac{n}{l}-2)}{2} = \frac{(n-l)\cdot(n-2l)}{2l}\ .$$

■

Theorem 4.4 *The algorithm approximates* Π*-Editing to within a factor of* td*.*

Proof: Let G^* be an optimally modified graph. If G^* contains no induced copy of any $T(F) \in \mathcal{F}'$ then by Theorem 4.1 algorithm APPROX achieves an approximation ratio of td. Otherwise, let H be an induced copy in G^* of some $T(F) \in \mathcal{F}'$. We first show that the size of an optimum Π-editing set for G is least $\frac{n-t(d+1)}{2}$.

Let $S \subseteq V \setminus V(H)$ denote the set of vertices which are not adjacent to any vertex of H. Since $|V(H)| \leq t-1$, $|S| \geq n-(t-1)(d+1) > n-t(d+1) \geq 2t(d+1)$.

Since G^* contains $T(F)$ but no induced copy of F and $|V(F)| - |V(T(F))| \leq t-2$, the size of a maximum independent set in G^*_S is at most $\max\{t-3, 1\} \leq t-2$. By Lemma 4.3 we conclude that $|E(G^*_S)| \geq \frac{(|S|-(t-2))\cdot(|S|-2(t-2))}{2(t-2)}$. But $|E(G_S)| \leq \frac{|S|d}{2}$. Hence,

$$\begin{aligned} |E(G^*)\triangle E| &\geq |E(G^*_S)| - \frac{|S|d}{2} > \frac{(|S|-(t-2))\cdot(2(t-2)(d+1)-2(t-2)) - d|S|(t-2)}{2(t-2)} = \\ & \frac{2d(|S|-(t-2)) - d|S|}{2} > \frac{d(|S|-2t)}{2} > \frac{|S|}{2} > \frac{n-t(d+1)}{2}\ . \end{aligned}$$

As algorithm APPROX only removes edges, the size of the editing set it produces is at most $|E| \leq \frac{nd}{2}$. Hence, the approximation ratio it achieves is at most $\frac{nd/2}{(n-t(d+1))/2} \leq \frac{nd}{2n/3} = \frac{3d}{2} < td$. ■

The last remaining case is when $\mathcal{F}$ contains an independent set as a forbidden induced subgraph. For the deletion problem, if $n \leq (d+1)t$ then it can be solved exactly in constant time by exhaustive search, and otherwise, it has no solution since a d-degree bounded graph with at least $(d+1)t$ vertices contains an independent set of size at least t.

We now handle the editing problem in this case. W.l.o.g. $n \geq 3(t-1)(d+1)$. Due to Ramsey's Theorem (cf. [6]), if $\mathcal{F}$ contains also a clique as a forbidden induced subgraph, then the number of vertices in G is bounded by a constant, and we can solve the problem exactly in constant time. Otherwise, we obtain a $3(t-1)$-approximation algorithm for the Π-Editing problem by simply transforming G into a clique. We prove this approximation ratio below.

Theorem 4.5 *The algorithm approximates* Π*-Editing to within a factor of* $3(t-1)$*.*

Proof: Let G be the input d-degree bounded graph. Let G^* be an optimally modified graph with maximum independent set of size at most $t-1$. By Lemma 4.3 G^* contains at least $\frac{(n-(t-1))\cdot(n-2(t-1))}{2(t-1)}$ edges. Thus, $|E(G^*)\triangle E(G)| \geq \frac{(n-(t-1))\cdot(n-2(t-1))}{2(t-1)} - \frac{nd}{2}$. Since the approximation algorithm adds at most $\binom{n}{2}$ edges, it achieves a ratio of at most

$$\frac{n(n-1)(t-1)}{(n-(t-1))\cdot(n-2(t-1))-nd(t-1)} \leq \frac{n(n-1)(t-1)}{\frac{2n}{3}(n-(t-1))-nd(t-1)} \leq$$
$$\frac{(n-1)(t-1)}{\frac{2}{3}(n-\frac{3}{2}(d+1)(t-1))} \leq \frac{(n-1)(t-1)}{\frac{n}{3}} < 3(t-1)\ .$$

The algorithm computes a correct solution since by our assumption $\mathcal{F}$ contains no clique as a forbidden induced subgraph. ■

The following theorem summarizes our results.

Theorem 4.6 *For any graph property Π which is characterized by a finite set of forbidden induced subgraphs, there is a polynomial time constant factor approximation algorithm for the Π-Deletion and Π-Editing problems on bounded degree graphs.*

4.2 Polynomial Algorithms

In the following we give polynomial algorithms for Chain Deletion and Editing, Split Deletion and Threshold Deletion and Editing when restricted to bounded degree graphs. These results are derived by observing that for these properties the search space becomes bounded when the problem is restricted to bounded degree graphs.

For the results concerning editing problems we need the following lemma.

Lemma 4.7 *Let Π be a hereditary graph property such that if $G=(V,E)$ satisfies Π then $G_{V\setminus\{v\}}\cup v$ satisfies Π for every $v\in V$ (i.e., the property remains satisfied if we remove all the edges incident on a vertex v). Then an optimum solution of Π-Editing on a d-degree bounded graph produces a graph with degree bounded by $2d$.*

Proof: The lemma follows by noting that it is never beneficial to add more than d edges incident on the same vertex, since one could instead make that vertex isolated by modifying fewer edges. ■

Theorem 4.8 *Chain Deletion and Chain Editing are polynomially solvable on bounded degree graphs.*

Proof: Let G be an input d-degree bounded graph. The proof follows from the observation that a chain graph with degree bounded by d has at most $2d$ vertices with degree at least one. Hence, a maximum chain subgraph of

G has at most $2d$ vertices with degree at least one. This set of vertices can be found by complete enumeration in polynomial time. Similarly, an optimum solution to the editing problem is by Lemma 4.7 a $2d$-degree bounded graph, and therefore has at most $4d$ vertices with degree at least one. ■

Theorem 4.9 *Split Deletion is polynomially solvable on bounded degree graphs.*

Proof: The proof follows from the observation that a split graph with degree bounded by d has maximum clique of size at most $d + 1$. Hence, one can enumerate all possible partitions of the vertex set of the graph into a clique and an independent set in polynomial time. ■

A graph $G = (V, E)$ is called a *threshold graph*, if there is a partition (K, I) of V such that K induces a clique, I induces an independent set, and the bipartite graph $(K, I, E \cap (K \times I))$ is a chain graph (cf. [29] for another equivalent definition of this class).

Theorem 4.10 *Threshold Deletion and Threshold Editing are polynomially solvable on bounded degree graphs.*

Proof: Let $G = (V, E)$ be an input d-degree bounded graph. An optimum threshold deletion set produces a graph with degree bounded by d. By Lemma 4.7, an optimum threshold editing set produces a graph with degree bounded by $2d$. Hence, one can enumerate all partitions of V into a clique and an independent set in polynomial time, and for each partition solve a chain modification problem on the corresponding bipartite graph using the result of Theorem 4.8. ■

5 Concluding Remarks

Most of the results obtained here and previously on edge modification problems are hardness results. Proving a general hardness result similar to that obtained for vertex deletion problems [26], is a challenging open problem.

The study of bounded-degree edge modification problems is still very preliminary. Such restriction is motivated by some real applications (see, e.g., [23]). Other realistic restrictions may be appropriate for particular problems. Studying the parameterized complexity of the NP-hard problems is also of interest.

Like every attempt to organize a body of results into a table or a diagram, Figure 1 immediately identifies numerous open problems. We conjecture that Chain Editing is NP-complete. If true, this would imply, among other results, the NP-completeness of Interval Editing and Unit Interval Editing.

References

[1] A. AGRAWAL, P. KLEIN, AND R. RAVI, *Cutting down on fill using nested dissection: provably good elimination orderings*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer, 1993, pp. 31–55.

[2] T. ASANO, *An application of duality to edge-deletion problems*, SIAM Journal on Computing, 16 (1987), pp. 312–331.

[3] T. ASANO AND T. HIRATA, *Edge-deletion and edge-contraction problems*, in Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, San Francisco, California, 5–7 May 1982, pp. 245–254.

[4] A. BEN-DOR. Private communication, 1996.

[5] H. BODLAENDER AND B. DE FLUITER, *On intervalizing k-colored graphs for DNA physical mapping*, Discrete Applied Math., 71 (1996), pp. 55–77.

[6] B. BOLLOBÁS, *Random Graphs*, Academic Press, 1985. Chapter XII.

[7] A. BRANDSTÄDT, V. B. LE, AND J. P. SPINRAD, *Graph Classes - a Survey*, SIAM, Philadelphia, 1999. SIAM Monographs in Discrete Mathematics and Applications.

[8] L. CAI, *Fixed-parameter tractability of graph modification problems for hereditary properties*, Information Processing Letters, 58 (1996), pp. 171–176.

[9] K. CIRINO, S. MUTHUKRISHNAN, N. NARAYANASWAMY, AND H. RAMESH, *Graph editing to bipartite interval graphs: exact and asymptotic bounds*, tech. report, Bell Laboratories Innovations, Lucent Technologies, 1996.

[10] D. G. CORNEIL, S. OLARIU, AND L. STEWART, *Asteroidal triple-free graphs*, SIAM Journal on Discrete Mathematics, 10 (1997), pp. 399–430.

[11] EL-MALLAH AND COLBOURN, *The complexity of some edge deletion problems*, IEEE Transactions on Circuits and Systems, 35(3) (1988), pp. 354–362.

[12] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, 1979.

[13] M. R. GAREY, D. S. JOHNSON, AND L. STOCKMEYER, Theor. Comput. Sci., 1 (1976), pp. 237–267.

[14] P. W. GOLDBERG, M. C. GOLUMBIC, H. KAPLAN, AND R. SHAMIR, *Four strikes against physical mapping of DNA*, Journal of Computational Biology, 2 (1995), pp. 139–152.

[15] M. GOLUMBIC, H. KAPLAN, AND R. SHAMIR, *Graph sandwich problems*, Journal of Algorithms, 19 (1995), pp. 449–473.

[16] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[17] M. C. GOLUMBIC, H. KAPLAN, AND R. SHAMIR, *On the complexity of DNA physical mapping*, Advances in Applied Mathematics, 15 (1994), pp. 251–261.

[18] M. C. GOLUMBIC AND R. SHAMIR, *Complexity and algorithms for reasoning about time: A graph-theoretic approach*, J. ACM, 40 (1993), pp. 1108–1133.

[19] S. L. HAKIMI, E. F. SCHMEICHEL, AND N. E. YOUNG, *Orienting graphs to optimize reachability*, Information Processing Letters, 63 (1997), pp. 229–235.

[20] P. L. HAMMER, T. IBARAKI, AND U. N. PELED, *Threshold numbers and threshold completions*, in Studies on Graphs and Discrete Programming, P. Hansen, ed., North-Holland, 1981, pp. 125–145.

[21] P. L. HAMMER AND B. SIMEONE, *The splittance of a graph*, Combinatorica, 1 (1981), pp. 275–284.

[22] J. HÅSTAD, *Some optimal inapproximability results*, in Proc. 29th STOC, 1997, pp. 1–10. Full version: E-CCC Report number TR97-037.

[23] H. KAPLAN AND R. SHAMIR, *Bounded degree interval sandwich problems*, Algorithmica, 24 (1999), pp. 96–104.

[24] H. KAPLAN, R. SHAMIR, AND R. E. TARJAN, *Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs*, SIAM Journal on Computing, 28 (1999), pp. 1906–1922.

[25] T. KASHIWABARA AND T. FUJISAWA, *An NP-complete problem on interval graphs*, in IEEE International Symposium on Circuits and Systems (12th), 1979, pp. 82–83.

[26] J. LEWIS AND M. YANNAKAKIS, *The node deletion problem for hereditary properties is NP-complete*, J. Comput. Sys. Sci., 20 (1980), pp. 219–230.

[27] L. LOVÁS, *A characterization of perfect graphs*, J. Combin. Theory, (1972), pp. 95–98.

[28] C. LUND AND M. YANNAKAKIS, *The approximation of maximum subgraph problems*, in Proceedings of International Conference on Automata, Languages and Programming (ICALP '91), A. Lingas, R. Karlsson, and S. Carlsson, eds., Berlin, Germany, 1993, Springer, pp. 40–51. LNCS 700.

[29] N. MAHADEV AND U. PELED, *Threshold Graphs and Related Topics*, Elsevier, North-Holland, 1995. Annals of Discrete Mathematics, Vol. 56.

[30] F. MARGOT, *Some complexity results about threshold graphs*, Discrete Applied Mathematics, 49 (1994).

[31] A. NATANZON, R. SHAMIR, AND R. SHARAN, *A polynomial approximation algorithm for the minimum fill-in problem*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC'98), New York, May 23–26 1998, ACM Press, pp. 41–47.

[32] A. NATANZON, R. SHAMIR, AND R. SHARAN, *Complexity classiffication of some edge modification problems*, in Proc. 25th Int. Workshop (WG '99), Graph-Theoretic Concepts in Computer Science, Springer-Verlag, 1999.

[33] R. RAVI, A. AGRAWAL, AND P. KLEIN, *Ordering problems approximated: single processor scheduling and interval graph completion*, in Proc. ICALP 1991, Springer, 1991, pp. 751–762. LNCS 510.

[34] J. D. ROSE, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. C. Reed, ed., Academic Press, N.Y., 1972, pp. 183–217.

[35] R. E. TARJAN AND M. YANNAKAKIS, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Computing, 13 (1984), pp. 566–579.

[36] L. TREVISAN, G. SORKIN, M. SUDAN, AND D. WILLIAMSON, *Gadgets, approximation, and linear programming*, in Proc. IEEE Symposium on Foundations of Computer Science (FOCS'96), 1996, pp. 617–626.

[37] J. XUE, *Edge-maximal triangulated subgraph and heuristics for the maximum clique problem*, tech. report, Graduate School of Management, Clark University, Worcester, MA, July 1993.

[38] M. YANNAKAKIS, *Computing the minimum fill-in is NP-complete*, SIAM J. Alg. Disc. Meth., 2 (1981).

[39] M. YANNAKAKIS, *Edge deletion problems*, SIAM Journal on Computing, 10(2) (1981), pp. 297–309.