Complexity Classification of Some Edge Modification Problems

Assaf Natanzon[1], Ron Shamir[1], and Roded Sharan[1]

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

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 in this paper 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.

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 family of graphs. 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 $G' = (V, E \triangle F) \in \Pi$, 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 edges, i.e., $F \cap E = \emptyset$. Equivalently, we seek a minimum supergraph of G with property Π. In this paper we study 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 [11, 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, 8, 12, 14]). 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 [30].

Chordal deletion problems occur when trying to solve the CLIQUE problem. Some heuristics for finding a large clique (see, e.g., [33]) 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 [24] showed that for any property which is non-trivial and hereditary, the maximum induced subgraph problem is NP-complete. Furthermore Lund and Yannakakis [26] 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 [3, 10, 2]. 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 in [34]. As noted in [12], the NP-completeness of Interval Completion and Unit Interval Completion also follows from [34]. Interval Completion was directly shown to be NP-complete in [11, problem GT35] and [23]. Deletion problems on interval graphs and unit interval graphs were proven to be NP-complete in [12]. Cograph Completion and Cograph Deletion were shown to be NP-complete in [10]. Threshold Completion and Threshold Deletion were shown to be NP-complete in [27]. Comparability Completion was shown to be NP-complete in [17] and Comparability Deletion was shown to be NP-complete in [35].

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 [8]. Split Editing was shown to be polynomial in [19].

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 [16, 15, 18, 21]. Several results on the parametric complexity of completion problems were also obtained [22, 7].

Approximation algorithms exist for several problems. In [28] 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 [29]. In [8] 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 w.r.t. any hereditary property that is 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 results on bounded degree graphs. For lack of space, some proofs are omitted and many corollaries are only alluded to in Figure 1.

2 Basic Results

In this section we summarize some easy observations on 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 $V(G)$. We denote by $\overline{G}$ the *complement graph* of G, i.e., $\overline{G} = (V, \bar{E})$, where $\bar{E} = (V \times V) \setminus E$. (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 $\overline{G} = (U, V, \bar{E})$, where $\bar{E} = (U \times V) \setminus E$. 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 a vertex $v \notin V$ we denote by $G \cup v$ the graph obtained by adding v to G as an isolated vertex. We denote by $G + v$ the graph obtained from G by adding v and connecting it to every other vertex of G. For a graph property Π the notation $G \in \Pi$ implies that G satisfied Π. For basic definitions of graph properties and much more on the graph classes discussed here see, e.g., [13, 6].

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* w.r.t. Π. *k-deletion set* and *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 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. *If Π is an ancestral graph property then Π-Completion and Π-Editing are polynomially equivalent, and Π-Deletion is not meaningful.*

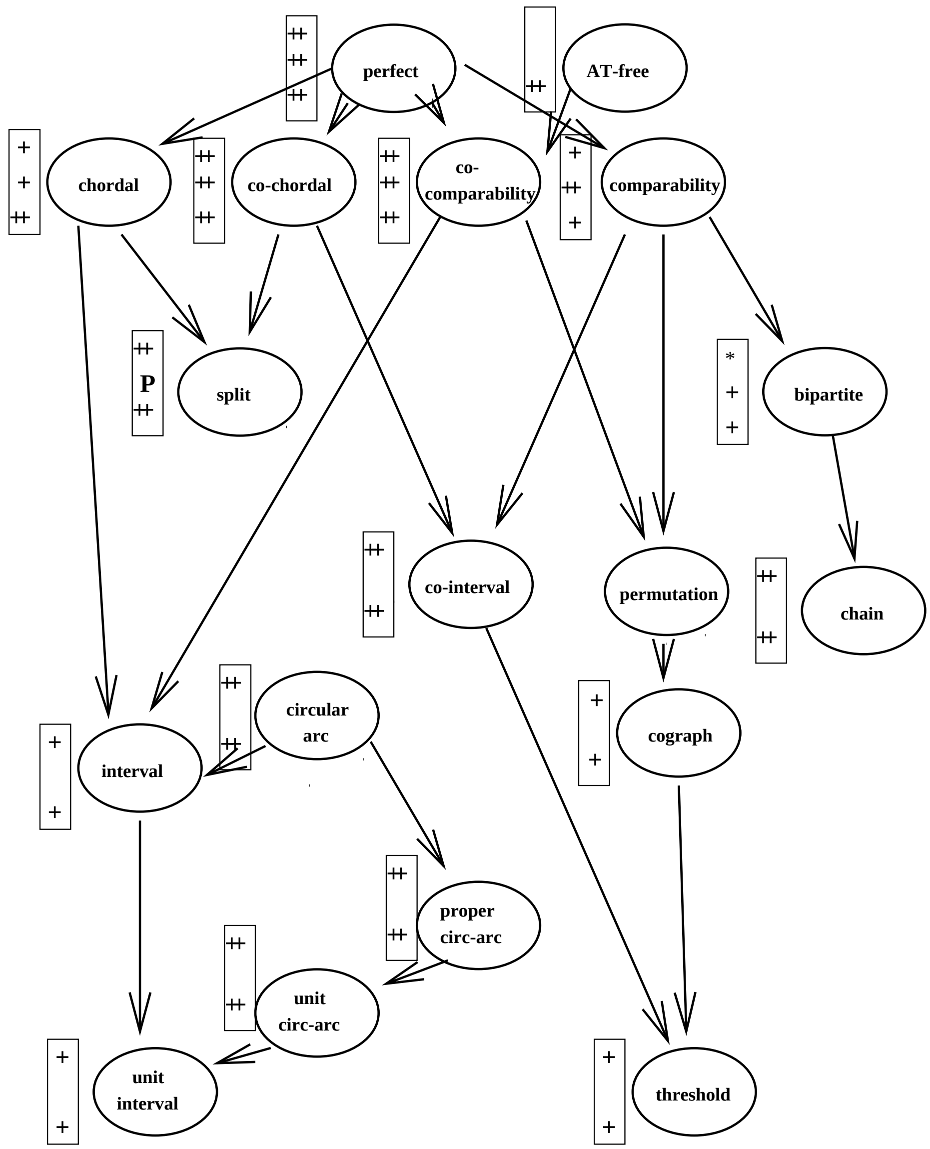

Fig. 1. The complexity status of edge modification problems for some graph classes. A→B indicates that class A contains class B. The box to the left of each class contains the status of the completion (top), editing (middle) and deletion (bottom) problems. $+$: NP-hard, previously known; $++$: NP-hard,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

Proposition 3. *If Π and Π' are graph properties such that for every graph G and a vertex $v \notin V(G)$, G satisfies Π iff $G \cup v$ satisfies Π', then Π-Deletion is polynomially reducible to Π'-Deletion. If in addition, Π' is a property such that $G \in \Pi'$ implies $G \cup v \in \Pi'$, then Π-Completion (Π-Editing) is polynomially reducible to Π'-Completion (Π'-Editing).*

Proposition 4. *If Π and Π' are graph properties such that for every graph G and a vertex $v \notin V(G)$, G satisfies Π iff $G + v$ satisfies Π', then Π-Completion is polynomially reducible to Π'-Completion. If in addition, Π' is a property such that $G \in \Pi'$ implies $G + v \in \Pi'$, then Π-Deletion (Π-Editing) is polynomially reducible to Π'-Deletion (Π'-Editing).*

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

Proposition 5. *For every graph property Π, Π-Deletion and $\overline{\Pi}$-Completion are polynomially equivalent.*

Proposition 6. *For every graph property Π, Π-Editing and $\overline{\Pi}$-Editing are polynomially equivalent.*

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 [34]. He also showed that G is a chain graph iff 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.

Lemma 1. *The bipartite complement of a chain graph is a chain graph.*

Proof. The claim follows from the observation that the ordering of containment is reversed for the bipartite complement of a chain graph.■

Corollary 1. *Chain Deletion is NP-complete.*

3.2 Perfect Graphs

A graph $G = (V, E)$ is called *perfect* if for any 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. It is easy to see that a perfect graph contains no induced cycle of odd length.

Theorem 1. *Perfect Completion is NP-hard.*

Proof: Reduction from Chain Completion. Let $< G = (P, Q, E), k >$ be an instance of Chain Completion. Since Chain Completion is NP-hard even when the input graph is connected [34], we can assume that the partition (P, Q) is known. 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

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

We now prove the 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 $K = (N, E' \cup F)$ and let $H = (V_H, E_H)$ be any induced subgraph of K. 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 $|V_1|$ colors as follows:
   (a) Color the vertices of V_1 with $|V_1|$ colors.
   (b) Color the vertices of $V_2 \cap Q$ with the color of p.
   (c) Color the vertices of type $v^2_{q_1,q_2,i}$ with the color of p.
   (d) Color the vertices of types $v^1_{q_1,q_2,i}$ and $v^3_{q_1,q_2,i}$ with any other color.

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

$\Leftarrow$ Suppose that F is a perfect k-completion set. 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 five 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. ■

The perfect graph theorem by Lovasz [25] states that the complement of a perfect graph is perfect. Hence, we conclude that Perfect Deletion is also NP-hard.

Theorem 2. *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

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

The validity proof is similar to that of Theorem 1 and is omitted. The additional edges of E_2 "protect" the edges in $E \cup (P \times P)$ and prevent their removal.■

3.3 Chordal Graphs

A graph is called *chordal* if it contains no induced cycle of length greater than 3. We show in this section that Chordal Deletion is NP-complete.

Theorem 3. *Chordal Deletion is NP-complete.*

Proof: The problem is in NP since chordal graphs can be recognized in linear time [31]. 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 iff the Chain Deletion instance has a solution.

- ⇒ 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 it must contain at least two vertices from P, a contradiction. The same holds for V_Q. Hence, $C \cap V_P = C \cap V_Q = \emptyset$. 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 $(P, Q, E \setminus F)$, a contradiction.
- ⇐ Suppose that F is a chordal k-deletion set. We will 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, both pairs are still connected in $H = (V', E' \setminus F)$ and p_1, q_1, q_2 and p_2 are on an induced cycle of length at least 4 in H, a contradiction. ■

3.4 Split graphs

A graph G is called a *split graph* if there is a partition (K, I) of $V(G)$, so that K induces a clique and I induces an independent set. 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 4. *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 w.r.t. 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.■

3.5 AT-Free Graphs

An *asteroidal triple* is a set of three independent vertices such that there is a path between every pair of vertices which avoids the neighborhood of the third vertex. G is called *Asteroidal Triple free*, or *AT-free*, if G contains no asteroidal triple. Several families of graphs are asteroidal triple free, e.g., interval and co-comparability graphs. For characterizations of AT-free graphs see cf. [9]. We prove here that AT-free Deletion is NP-complete.

Theorem 5. *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 $< (V', E'), k >$ of AT-free Deletion: Define $V' = U \cup V \cup V_q \cup V_w \cup V_z$, where $V_q = \{q_1, \ldots, q_k\}, V_w = \{w_1, \ldots, w_{k+1}\}$ and $V_z = \{z_1, \ldots, z_{k+1}\}$. Define $E' = E \cup (U \times U) \cup (U \times V_q) \cup (U \times V_w) \cup ((V_w \cup V_z) \times (V_w \cup V_z))$. The validity proof is omitted.■

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 $(a, c) \in F$. 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 6. *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

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

(for each $(v, w) \in E$ the choice of which vertex to connect to $e^1_{v,w}$ is arbitrary). In other 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 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 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 $v \in V_2$ into v. 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 $e^1_{v,w}$.
- ⇐ Suppose that F is a solution to the comparability instance, 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. Define a partition (V_1, V_2) of V, in which $v \in V_1$ iff 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 is incident on any of those paths. For each such path, its corresponding edge must be across the cut, as otherwise R could not have been transitive. ■

A slight modification of the above reduction shows that if Comparability Editing can be approximated with ratio $1 + \theta$ then MAX-CUT can be approximated with ratio $1/(1 - \theta)$. In [32, 20] it is shown that approximating MAX-CUT to within a factor of $17/16$ is NP-hard. We conclude:

Corollary 2. *It is NP-hard to approximate Comparability Editing to within a factor of* $18/17$.

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

We present below a constant factor approximation algorithm for the deletion and editing problems on bounded degree graphs. The result applies to any hereditary family which can be characterized by a finite set of forbidden induced subgraphs. Examples include cographs and claw-free graphs. An analogous result for vertex deletion problems was given by Yannakakis and Lund [26]. We also show that for

bounded degree graphs Chain Deletion and Editing, Split Deletion and Threshold Deletion and Editing are polynomial.
Let Π be an 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. In the following we further assume that no forbidden subgraph contains an isolated vertex. The approximation algorithm follows.

1) $A \leftarrow \emptyset$
2) **While** $G_{V \setminus A}$ contains an induced subgraph H isomorphic to some $F \in \mathcal{F}$, **do**:
   $A \leftarrow A \cup V(H)$.
3) 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 7. *The algorithm approximates Π-Deletion and Π-Editing to within a factor of td.*

Proof. **Correctness**: After Step 2 is completed, $G_{V \setminus A}$ contains no forbidden induced subgraph. After Step 3 is completed, all vertices in A become isolated. Since no forbidden induced subgraph contains an isolated vertex, at the end of the algorithm $G \in \Pi$.
Approximation ratio: Let F be an optimum solution of size k. For any forbidden induced subgraph H found at Step 2 of the algorithm, F must contain an edge incident on H. Hence, at the end of the algorithm $|A| \leq kt$, and at most ktd edges are deleted from G.■

It can be shown that our result extends for all hereditary properties that can be characterized by a finite set of forbidden induced subgraphs.
In the following we give some polynomial results for edge modification problems on bounded degree graphs. These results are derived by observing that for the properties in question the search space becomes bounded when the problem is restricted to bounded degree graphs.

Theorem 8. *Chain Deletion and Chain Editing are polynomially solvable on bounded degree graphs.*

Theorem 9. *Split Deletion is polynomially solvable on bounded degree graphs.*

Theorem 10. *Threshold Deletion and Threshold Editing are polynomially solvable on bounded degree graphs.*

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 [24], 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., [21]).

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. Many of those have not been investigated yet, and we are in the process of studying some of them.

References

1. A. Aggarwal, P. Klein, and R. Ravi. Cutting down on fill using nested dissection: provably good elimination orderings. In A. George, J. R. Gilbert, and J. W. H. Liu, editors, *Graph Theory and Sparse Matrix Computation*, pages 31–55. Springer, 1993.
2. T. Asano. An application of duality to edge-deletion problems. *SIAM Journal on Computing*, 16(2):312–331, 1987.
3. T. Asano and T. Hirata. Edge-deletion and edge-contraction problems. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 245–254, San Francisco, California, 5–7 May 1982.
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.*, 71:55–77, 1996.
6. A. Brandstädt and J. Spinrad. Graph classes - a survey. manuscript, 1997.
7. L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58:171–176, 1996.
8. K. Cirino, S. Muthukrishnan, N. Narayanaswamy, and H. Ramesh. Graph editing to bipartite interval graphs: exact and asymptotic bounds. Technical report, Bell Laboratories Innovations, Lucent Technologies, 1996.
9. D. G. Corneil, S. Olariu, and L. Stewart. The linear structure of graphs: asteroidal triple-free graphs. In *Proc. 19th Int. Workshop (WG '93), Graph-Theoretic Concepts in Computer Science*, pages 211–224. Springer-Verlag, 1994. LNCS 790.
10. El-Mallah and Colbourn. The complexity of some edge deletion problems. *IEEETCS: IEEE Transactions on Circuits and Systems*, 35(3):354–362, 1988.
11. 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.
12. P. W. Goldberg, M. C. Golumbic, H. Kaplan, and R. Shamir. Four strikes against physical mapping of DNA. *Journal of Computational Biology*, 2(1):139–152, 1995.
13. M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
14. M. C. Golumbic, H. Kaplan, and R. Shamir. On the complexity of DNA physical mapping. *Advances in Applied Mathematics*, 15:251–261, 1994.
15. M. C. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. *Journal of Algorithms*, 19:449–473, 1995.
16. M. C. Golumbic and R. Shamir. Complexity and algorithms for reasoning about time: A graph-theoretic approach. *J. ACM*, 40:1108–1133, 1993.
17. S. L. Hakiri and N. R. Young. Orienting graphs to optimize reachability. *Information Proc. Letters*, 1997.

18. P. L. Hammer, T. Ibaraki, and U. N. Peled. Threshold numbers and threshold completions. In P. Hansen, editor, *Studies on Graphs and Discrete Programming*, pages 125–145. North-Holland, 1981.
19. P. L. Hammer and B. Simeone. The splittance of a graph. *Combinatorica*, 1:275–284, 1981.
20. J. Håstad. Some optimal inapproximability results. In *Proc. 29th STOC*, pages 1–10, 1997. Full version: E-CCC Report number TR97-037.
21. H. Kaplan and R. Shamir. Physical maps and interval sandwich problems: Bounded degrees help. In *Proc. ISTCS*, pages 195–201, 1996.
22. H. Kaplan, R. Shamir, and R. E. Tarjan. Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping. In *Proceedings of the 35th Symposium on Foundations of Computer Science*, pages 780–791. IEEE Computer Science Press, Los Alamitos, California, 1994. to appear in *SIAM J. Computing*.
23. T. Kashiwabara and T. Fujisawa. An NP-complete problem on interval graphs. In *IEEE International Symposium on Circuits and Systems (12th)*, pages 82–83, 1979.
24. J. Lewis and M. Yannakakis. The node deletion problem for hereditary properties is NP-complete. *J. Comput. Sys. Sci.*, 20:219–230, 1980.
25. L. Lovás. A characterization of perfect graphs. *J. Combin. Theory*, pages 95–98, 1972.
26. C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In A. Lingas, R. Karlsson, and S. Carlsson, editors, *Proceedings of International Conference on Automata, Languages and Programming (ICALP '91)*, pages 40–51, Berlin, Germany, 1993. Springer. LNCS 700.
27. F. Margot. Some complexity results about threshold graphs. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 49, 1994.
28. 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)*, pages 41–47, New York, May 23–26 1998. ACM Press.
29. R. Ravi, A. Agrawal, and P. Klein. Ordering problems approximated: single processor scheduling and interval graph completion. In *Proc. ICALP 1991*, pages 751–762. Springer, 1991. LNCS 510.
30. J. D. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In R. C. Reed, editor, *Graph Theory and Computing*, pages 183–217. Academic Press, N.Y., 1972.
31. R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, text acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Computing*, 13:566–579, 1984.
32. 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)*, pages 617–626, 1996.
33. J. Xue. Edge-maximal triangulated subgraph and heuristics for the maximum clique problem. Technical report, Graduate School of Management, Clark University, Worcester, MA, July 1993.
34. M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2, 1981.
35. M. Yannakakis. Edge deletion problems. *SIAM J. Computing*, 10(2):297–309, 1981.