Tel Aviv University

The Raymond and Beverly Sackler

Faculty of Exact Sciences

School of Mathematical Sciences

Complexity and Approximation of Some Graph Modification Problems

Thesis submitted in partial fulfillment

of graduate requirements for the degree

M.Sc. in Tel Aviv University

Department of Computer Science

By

Assaf Natanzon

Prepared under the supervision of Prof. Ron Shamir

Acknowledgments

I would like to thank my advisor Prof. Ron Shamir, for directing me and giving me a helping hand, and for his clarifying comments and reformulations.

I would like to thank my beloved Mom and Dad, for their support.

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, comparability, split, threshold and cluster graphs. We show that some of these problems become polynomial when the input graph has bounded degree. We 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. We give an approximation algorithm for Cluster Deletion and Editing. We also prove some combinatorial bounds on the sizes of the minimum completion and deletion sets of a graph.

Contents

Chapter 1

Introduction and Summary

1.1 Introduction

In this thesis we deal with graph modification problems. Modification problems on graphs play an important role in computer science and have applications in several fields, including computer algebra and molecular biology. They include *edge completion* problems, *edge deletion* problems, *edge editing* problems and *vertex deletion* problems.

We now define the generic problems we shall study in this thesis. All problems are defined below with respect to a family of graphs $\mathcal{F}$ that is not a part of the input. For example, $\mathcal{F}$ can be the set of chordal graphs or the perfect graphs, etc.

The Minimum Completion problem is defined as follows: given an arbitrary graph $G = (V, E)$, find a set of non-edges F such that $|F|$ is minimum, and $G' = (V, E+F) \in \mathcal{F}$, *i.e.,* find a minimum set of edges whose addition to G will form a graph that belongs to $\mathcal{F}$. For a graph $G = (V, E)$, a *k-completion set for a graph family $\mathcal{F}$* (or an *$\mathcal{F}$-k-completion*) is a set F of at most k edges so that $(V, E + F) \in \mathcal{F}$.

The Minimum Deletion problem is defined as follows: given an arbitrary graph

$G = (V, E)$, find a set of edges H such that $|H|$ is minimum, and $G' = (V, E \setminus H) \in \mathcal{F}$, *i.e.,* find a minimum set of edges F whose removal from G will form a graph that belongs to $\mathcal{F}$. For a graph $G = (V, E)$, a *k-deletion set for a graph family* $\mathcal{F}$ (or an $\mathcal{F}$*-k-deletion*) is a set F of at most k edges so that $(V, E \setminus F) \in \mathcal{F}$.

The Minimum Editing problem is defined as follows: given an arbitrary graph $G = (V, E)$, find a set of non-edges F and a set of edges H, such that $|F \cup H|$ is minimum, and $G' = (V, (E + F) \setminus H) \in \mathcal{F}$, *i.e.,* find a minimum set of edges you need to edit (add or delete) such that editing of these edges in the graph G will form a graph that belongs to $\mathcal{F}$. For a graph $G = (V, E)$, a *k-editing set for a graph family* $\mathcal{F}$ (or an $\mathcal{F}$*-k-editing*) is a set $K = F \cup H$ of at most k edges so that $(V, (E+F) \setminus H) \in \mathcal{F}$.

The Minimum Vertex Deletion problem is defined as follows: given an arbitrary graph $G = (V, E)$, find a set of vertices F such that $|F|$ is minimum and the induced subgraph $G_{V \setminus F} = (V \setminus F, E \cap [(V \setminus F) \times (V \setminus F)])$ belongs to $\mathcal{F}$.

In the *Maximum Induced Subgraph Problem* one wishes to maximize $|V \setminus F|$ rather then to minimize $|F|$. The optimum solutions for both problems give the same graph, but they differ when approximation is sought.

In the sequel, we shall abbreviate "minimum completion problem for property $\mathcal{F}$" to $\mathcal{F}$-Completion. Names of other problems will be abbreviated in the same manner. We shall study, for example, Chordal Completion, Perfect Editing and Chain Deletion.

The importance of edge modification problems is both theoretical and practical. From the theoretical point of view, edge modification problems are natural problems in graph theory. Moreover, several fundamental problems in graph theory, such as maximum matching and max-cut, can be formulated as edge modification problems.

From the practical point of view, edge modification problems have important applications. Chordal Completion, which is also called the *minimum fill-in problem*, arises in numerically performing a Gaussian elimination on a sparse symmetric matrix, such that a minimum number of new non-zero elements is introduced (see [37, 42]).

Modification problems with respect to interval and proper interval graphs have application in physical mapping of DNA [16, 18]. Cluster graph modification problems arise in numerous application areas where clustering is called for (see, e.g. [5]) . In general, modification problems arise when the input data that is supposed to be represented by an $\mathcal{F}$-graph contains errors. False negative errors correspond to missing edges that should be added, and false positive errors correspond to edges that must be removed. One seeks a solution minimizing the number of errors, allowing one or both kinds of errors.

1.2 Definitions

All graphs in this study are finite and without parallel edges and self loops. Let $G = (V, E)$ denote a graph. For $w \in V$ define $Adj(w) = \{x|(w, x) \in E\}$. Given a subset $A \subseteq V$ of the vertices, we define the *subgraph induced* by A to be $G_A = (A, E_A)$, where $E_A = \{(x, y) \in E|x, y \in A\}$.

A *subgraph* of G is any graph (A, E') obtained by taking $A \subseteq V$ and $E' \subseteq E_A$. If G is a subgraph of H then we shall also say that H is a supergraph of G.

The *complement* of G is the graph $\bar{G} = (V, \bar{E})$, $\bar{E} = \{(x, y) \in V \times V|x \neq y$ and $(x, y) \notin E\}$.

A *clique* in a graph G is a complete induced subgraph. K_l is a clique graph of size l. A clique is *maximum* if there is no clique of G of larger cardinality. $\omega(G)$ is the number of vertices in a maximum clique of G.

A *stable* set or an *independent* set in a graph is a subset of vertices no two of which are adjacent.

A (proper) *c-coloring* of a graph is a partition of its vertices $V = X_1 + X_2 + ... + X_c$ such that each X_i is an independent set. The *chromatic number* of G, denoted $\chi(G)$, is the smallest possible c for which there exists a proper c-coloring of G.

A graph G is *bipartite* if G has a 2-coloring, *i.e.,* its vertices can be partitioned into two independent sets.

A sequence of vertices $[v_0, ..., v_k]$ is called a *path* of length k, if $(v_{i-1}, v_i) \in E$ for $i = 1, .., k$. A path $[v_0, ..., v_k]$ is called *simple* if $v_i \neq v_j$ for $i \neq j$. A simple path $[v_0, ..., v_k]$ is called *chordless* if $(v_i, v_j) \in E \Rightarrow |i - j| = 1$. P_k denotes a chordless path on k vertices.

A sequence of vertices $[v_0, ..., v_l]$ is called a *cycle* of length $l + 1$ if $(v_{i-1}, v_i) \in E$ for $i = 1, .., l$ and $(v_l, v_0) \in E$. A cycle $[v_0, ..., v_l]$ is called *simple* if $v_i \neq v_j$ for $i \neq j$. A simple cycle $[v_0, ..., v_l]$ is *chordless* if $(v_i, v_j) \in E \Rightarrow (|i - j| = 1$ or $|i - j| = l)$. C_k denotes a k-vertex chordless cycle.

For a graph G and an integer l, $l * G$ or lG is the graph formed by l independent copies of the graph G.

Let H be a graph. A graph G is called $H - free$ if G does not contain any induced copy of H. For example: a graph is a clique iff it is $2 * K_1 - free$. For a family $\mathcal{F}$ of graphs, we say that a graph G is $\mathcal{F}$ *– free* if it is $F - free$ for every $F \in \mathcal{F}$.

A family of graphs $\mathcal{F}$ is *hereditary* if for any graph $G \in \mathcal{F}$ every induced subgraph of G belongs to $\mathcal{F}$.

A graph $G = (V, E)$ is *perfect* if $\omega(G_A) = \chi(G_A)$ for every $A \subseteq V$.

A graph $G = (V, E)$ is *Berge* if it contains no induced chordless cycle of odd size and no induced complement of a chordless cycle of odd size.

We will be mainly interested in some graph subclasses of perfect graphs classes or in closely related classes. (For a survey on perfect graphs and for more information about these properties see [17]).

A bipartite graph $G = (P, Q, E)$ is called a *chain graph* if the vertices of one of the parts, say P, can be ordered v_1, v_2,..,v_k so that $Adj(v_1) \supseteq Adj(v_2) \supseteq ... \supseteq Adj(v_k)$, (recall that $Adj(v) = \{x | (v, x) \in E\}$). This property is called the *chain property.*

This property is hereditary, as if we remove a vertex from P or Q, the chain of containments does not change. For an example of a chain graph see Figure 1.1.

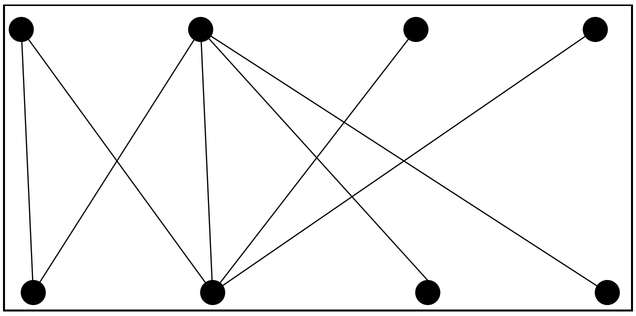

Figure 1.1: a chain graph

An undirected graph G is called *chordal* or *triangulated* if it contains no induced chordless cycle of length greater than three.

An *orientation* of an undirected graph $G = (V, E)$ is an assignment of a single directed arc uv or vu to each undirected edge $(u, v) \in E$. The resulting directed graph (V, F) is also called an orientation of G.

An undirected graph $G = (V, E)$ is a *comparability graph*, or *transitively orientable*, if there exists an orientation (V, F) of G satisfying: if $ab, bc \in F$ then $ac \in F$. In that case F is also called a *transitive orientation* of G.

A graph is called a *complement reducible* graph, or a *cograph*, if it contains no induced P_4 (a path with four vertices).

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

A graph $G = (V, E)$ is called a *threshold graph* if there is a partition (V_1, V_2) of V such that G_{V_1} is a clique, G_{V_2} is an independent set and the vertices of V_2 can be ordered w_1, w_2,..,w_k so that $Adj(w_1) \supseteq Adj(w_2) \supseteq ... \supseteq Adj(w_k)$.

A graph G is called a *cluster graph* if it is a disjoint union of cliques. The cluster property is hereditary.

A family of graphs $\mathcal{F}$ has a *forbidden set characterization* if there is a set $\mathcal{H}$ of graphs such that a graph is in $\mathcal{F}$ iff it does not contain any graph of $\mathcal{H}$ as an induced subgraph. $\mathcal{F}$ has a *finite forbidden set characterization* if $\mathcal{H}$ is finite. It is easy to see that a family of graphs $\mathcal{F}$ is hereditary iff it has a forbidden set characterization.

A graph property is *non trivial* if it is true for infinitely many graphs and false for infinitely many graphs.

1.3 Previous results

Yannakakis and Lewis showed that the vertex deletion problem for any non trivial hereditary family of graphs is NP-hard [30]. Furthermore Lund and Yannakakis [32] proved that for any non-trivial hereditary 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 $\tilde{P} = \tilde{NP}$.

Most of the results obtained so far, concerning edge modification problems, are NP-hardness ones. However, unlike the situation for vertex deletion [30], no general hardness results for edge modification is known.

In [42] Yannakakis solved a central open problem dealing with the complexity of Chordal Completion. He showed that this problem is NP-complete using a reduction from Chain Completion. The latter problem was proved to be NP-complete using a reduction from the optimal linear arrangement problem. As noted in [16] his proof implies that Interval Completion and Unit Interval Completion are also NP-complete. Interval Completion was directly shown to be NP-complete by Garey et al. (see problem GT35 in [14]) and by Kashiwabara and Fujisawa [29]. Deletion problems on interval graphs and unit interval graphs were proved to be NP-complete by Goldberg et al. [16]. Threshold Completion and Deletion were shown to be NP-complete in [33]. The hardness of Comparability Completion and Deletion was shown in [41] and [21] respectively.

Variants of the completion problem, in which the input graph is pre-colored and the objective is to find a supergraph satisfying a specified property, such that it is properly colored by the input coloring, were also shown to be NP-complete. Goldberg et al. [16] proved that Colored Unit Interval Completion is NP-complete. Golumbic et al. [18] and Fellows et al. [13] proved independently that Colored Interval Completion is NP-complete. Bodlaender and de Fluiter [7] strengthened this result by showing that the latter problem is NP-complete even if the number of colors is at most 4. They also gave a quadratic algorithm (in the number of vertices) for solving the Colored Interval Completion problem on 3-colored graphs [7]. Colored Chordal Completion was proved by Bodlaender et al. [6] to be NP-complete. McMorris et al. [34] showed that this problem is polynomial when the number of colors is fixed.

A generalization of the colored graph completion problems is to find a supergraph satisfying a given property which does not include forbidden edges (which are pre-defined). Problems of this type are called *sandwich* problems. Golumbic and Shamir [20] proved that the interval sandwich problem is NP-complete. Their proof was modified in [18] to prove that the unit interval sandwich problem is also NP-complete. Golumbic et al. [19] showed that sandwich problems for chordal graphs, comparability graphs, permutation graphs, circular-arc graphs, and several other families of graphs, are NP-complete. They also proved that the sandwich problem is polynomial for split graphs, threshold graphs (this was first shown by Hammer et al. [22]) and other families of graphs.

As for editing problems, the following results were obtained: Chordal Editing was proved to be NP-complete by Ben-Dor [4], using a reduction from Chordal Completion; Connected Bipartite Interval (*Caterpillar*) Editing was proved to be NP-complete by Cirino et al. [12]; and Split Editing was solved in polynomial time by Hammer and Simeone [23].

Since most edge modification problems discussed above are NP-complete, it is natural to investigate their parametric complexity. Kaplan and Shamir [25] gave a polynomial algorithm for the interval sandwich decision problem restricted to bounded degree input graphs, whenever the solution has bounded clique size or bounded degree. The results in [7] however, imply that the problem of finding an interval sandwich graph with a small clique is hard in the parametric sense, if the parameter is the size of the clique. In [26] Kaplan et al. proved that Chordal Completion and Unit Interval Completion are fixed parameter tractable, where the parameter is the number of edges added. The problem of altering a graph to one having a specified property, by deleting at most i vertices, deleting at most j edges, and adding at most k edges, where i, j, k are fixed integers, was proved by Cai [10] to be fixed parameter tractable for any hereditary property that has a finite forbidden set characterization.

Approximation results for edge modification problems are few. In [12] it was proved that the minimum number of edit operations needed to convert a graph to a caterpillar cannot be approximated in polynomial time to within an additive term of $O(n^{1-\epsilon})$, for $0 < \epsilon < 1$, unless $P = NP$. For Chordal Completion, Agrawal et al. [2] gave a polynomial algorithm that guarantees an approximation factor of $O(\sqrt{d}\log^4 |V|)$ for the case that the degree in G is bounded by d. They also gave an $O(|E|^{1/4}\log^{3.5}|V|)$ factor polynomial approximation algorithm for the general case. For Interval Completion a polynomial $O(\log^2 |V|)$ approximation algorithm was given by Agrawal et al. in [1]. These algorithms approximate the total number of edges in the supergraph containing G and not the number of modifications.

A polynomial algorithm for Chordal Completion was recently given by Natanzon et al. [35]. The algorithm constructs a triangulation whose size is at most a constant times the optimum size squared for the general case, and an $O(d^{2.5}log^4(kd))$ factor for the bounded degree case. A polynomial approximation algorithm achieving an $O(opt)$ ratio is also given in [35] for Chain Completion.

1.4 Summary of thesis results

The thesis has four parts. In the first part we show NP-hardness of some problems. The results of this part, as well as related results from the literature, are summarized in the following table;

graph family problem type	deletion	completion	editing
chain graphs	NP-complete [38]	NP-complete [42]	open
chordal graphs	NP-complete [38]	NP-complete [42]	NP-complete [4]
perfect graphs	NP-hard (new)	NP-hard (new)	NP-hard (new)
Berge graphs	NP-hard (new)	NP-hard (new)	NP-hard (new)
comparability graphs	NP-complete [21] *	NP-complete [42] *	NP-complete (new)*
split graphs	NP-complete(new)	NP-complete (new)	open
cographs	NP-complete(new)	NP-complete (new)	open
cluster graphs	NP-complete (new)	polynomial (trivial)	open

(*) Also: NP-hard to approximate to within a constant factor(new).

In the second part we give some approximation algorithms. We give a constant factor algorithm for the edge deletion/editing problem for families of graph with finite forbidden set characterization, on graphs with bounded degrees. We also give an $O(opt)$ approximation for the Cluster Editing/Deletion problem.

In the third part we give polynomial time algorithms for Chain Editing and Threshold Editing for bounded degree graphs.

In the fourth part we prove some combinatorial bounds on the sizes of the completion and deletion of a graph. We give a simple example of a graph with n vertices and $O(n^{1.5})$ edges whose smallest chordal supergraph contains $O(n^2)$ edges. We show that the smallest chordal supergraph containing a random graph $G(n, \frac{1}{\sqrt{n}})$ has $\Omega(\frac{n^2}{log^2(n)})$

edges and the smallest perfect graph containing a random graph $G(n, \frac{1}{n^{\frac{3}{5}}})$ has $\Omega(n^{\frac{9}{5}})$ edges. We also show that the minimum perfect deletion set of a random graph $G(n, \frac{1}{2})$ has $\Omega(n^2)$ edges.

1.5 Outline of the thesis

In chapter 2 we deal with NP-hardness of problems. Chapter 3 describes the approximation algorithms. In chapter 4 we give the polynomial time algorithms and chapter 5 deals with the combinatorial results.

Chapter 2

NP-Hardness results

In this chapter we give hardness results for various graph modification problems. We shall omit in the proofs for the arguments for membership in NP and the polynomiality of the reductions as they are standard.

2.1 Chain and Chordal Deletion

The chordal and chain deletion results were first obtained by Sharan [38, 36]. They are repeated for completeness, as the chain results are the basis to many of our subsequent reductions.

We need two simple characterizations of chain graphs.

Claim 2.1 *[42] If a bipartite graph is a chain graph, both parts of the graph have the chain property.*

Two edges (i, j), (k, l) are *independent* if the graph induced on $\{i, j, k, l\}$ is $2K_2$. (see Figure 2.1)

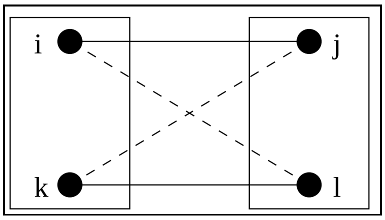

Figure 2.1: (i, j) and (k, l) are independent edges. Solid edges are present, broken lines denote missing edges.

Theorem 2.2 *[42] A bipartite graph is a chain graph iff it does not contain two independent edges.*

Theorem 2.3 *(Yannakakis, 1981 [42]) Chain Completion is NP-complete.*

For a bipartite graph $G = (P, Q, E)$ we define its *complement bipartite graph* by $\bar{G} = (P, Q, (P \times Q) \setminus E)$.

Lemma 2.4 *A bipartite graph is a chain graph iff its complement bipartite graph is a chain graph.*

Proof. If $G = (P, Q, E)$ is a chain graph, the vertices in P can be ordered $p_1, ..., p_k$ so that:

$$Adj(p_1) \supseteq Adj(p_2) \supseteq ... \supseteq Adj(p_k)$$

but this is true iff the complement bipartite graph $\bar{G}$ is a chain graph, since

$$Adj(p_k) \supseteq Adj(p_{k-1}) \supseteq ... \supseteq Adj(p_1)$$

in $\bar{G}$. ■

Lemma 2.5 *Chain Deletion is NP-complete.*

Proof. Reduction from Chain Completion : Given a bipartite graph $G = (P, Q, E)$ and a constant k, we will form its complement bipartite graph $\bar{G}$ and use same constant k. Let F be a set of edges we remove from $\bar{G}$ giving the graph $G' = (P, Q, (P \times Q) \setminus (E \cup F))$. By the previous lemma G' is a chain graph if and only if $\hat{G} = (P, Q, E \cup F)$ is a chain graph. Therefore F is a chain k-completion set for G iff it is a chain k-deletion set for $\bar{G}$. ■

For a bipartite graph $G = (P, Q, E)$ we define a new graph $C(G) = (N, E')$ where $N = P \cup P' \cup Q \cup Q'$ where P', Q' are sets of $k+1$ new vertices, and $E' = E + \{(u, v) | u, v \in P \cup P'\} + \{(u, v) | u, v \in Q \cup Q'\}$.

Lemma 2.6 *G is a chain graph iff $C(G)$ is triangulated.*

We omit the proof here. A proof of a similar lemma will be given in a next chapter.

Theorem 2.7 *Chordal Deletion is NP-complete.*

Proof. Reduction from Chain Deletion: Given a bipartite graph $G = (P, Q, E)$ and a constant k, we build the graph $C(G)$ and give the same constant k. By the previous lemma, if the deletion of a set of edges from G forms a chain graph, then deletion of the same set of edges in $C(G)$ gives a chordal graph.

Conversely, suppose F is a chordal k-deletion set in $C(G)$, and suppose $G' = (P, Q, E \setminus F)$ is not a chain graph: This means that G' contains two independent edges, namely there are $p_1, p_2 \in P$, $q_1, q_2 \in Q$ s.t. $(p_1, q_1), (p_2, q_2) \in E \setminus F$ and $(p_1, q_2), (p_2, q_1) \notin E \setminus F$. There are $k+1$ edge-disjoint paths between p_1 and p_2 passing through P'. Hence, after deleting F there is still a vertex $\hat{p} \in \hat{P}$ connected to p_1 and p_2. Similarly, there is still a vertex $\hat{q} \in \hat{Q}$ connected to q_1 and q_2. The cycle $p_1, \hat{p_1}, p_2, q_2, \hat{q}, q_1$ contains an induced cycle of length at least four, a contradiction. ■

2.2 Perfect Modifications

In this chapter we prove that completion, deletion and editing are all hard for perfect graphs. We need a key property of perfect graphs:

Theorem 2.8 *(The perfect graph theorem) (Lovasz, 1972 [31]) A graph is perfect if and only if its complement is perfect.*

We will also use the simple fact that for any odd $k > 1$ C_k and $\bar{C}_k$ are not perfect.

2.2.1 Perfect Completion

Theorem 2.9 *Perfect Completion is NP-Hard.*

Proof. Reduction from Chain Completion: The input is a bipartite graph $G = (P, Q, E)$ and a constant k. We use the same constant k and form the following graph: $P(G) = (N, E')$, $N = P \cup Q \cup C$ where

$$C = \{v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i} | q_1, q_2 \in Q, q_1 \neq q_2, 1 \leq i \leq k+1\}$$

and $E' = E \cup E_1 \cup E_2$ where

$$E_1 = \{(u,v)|u,v \in P\}$$

and

$$E_2 = \{(q_1, v^1_{q_1,q_2,i}), (v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i})(v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}), (v^3_{q_1,q_2,i}, q_2)|q_1, q_2 \in Q, 1 \leq i \leq k+1\}.$$

For an example of a graph produced by the reduction see Figure 2.2.

Lemma 2.10 *If G is a chain graph then $P(G)$ is a perfect graph.*

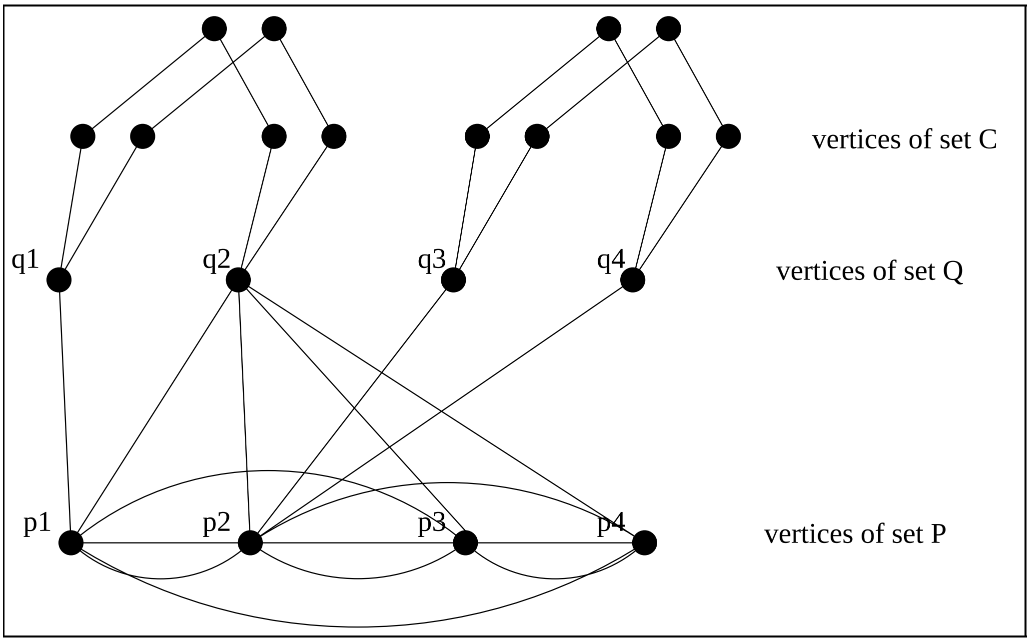

Figure 2.2: A part of the graph $P(G)$ in the proof of Theorem 2.9

Proof. Let $H = (V, \hat{E})$ be an induced subgraph of $P(G)$. We will show that $\omega(H) = \chi(H)$, *i.e.*, the coloring number of H is equal to the size of the maximum clique in H. Let $V = V_1 \cup V_2$ such that $V_1 \subseteq P$ and $V_2 \subseteq Q \cup C$. We distinguish two cases:

Case 1: If there is a vertex in V_2 which is connected to all the vertices in V_1, the clique number of H is at least $|V_1| + 1$ (this vertex and the vertices of V_1 form a clique).

We can color H with $|V_1| + 1$ colors in the following way:

1. Color the vertices of V_1 with $|V_1|$ colors.
2. Color the vertices that belong to Q with color number $|V_1| + 1$.
3. Color all the vertices of type $v^2_{q_1,q_2,i}$ with color number $|V_1| + 1$
4. Color all the vertices of types $v^1_{q_1,q_2,i}$, $v^3_{q_1,q_2,i}$ with color number $|V_1|$.

This is a vertex coloring of H with $|V_1|+1$ colors, therefore, $\chi(H) \leq \omega(H)$, but since for every graph G $\omega(G) \leq \chi(G)$, we have $\chi(H) = \omega(H)$.

Case 2: If no vertex in V_2 is connected to all vertices in V_1, the maximum clique size of H is at least $|V_1|$. Since G is a chain graph there is an order $q_1, q_2, .., q_k$ of the vertices of $V_2 \cap Q$ such that $Adj(q_1) \supseteq Adj(q_2) \supseteq ... \supseteq Adj(q_k)$. This means that there is a vertex p in V_1 such that no vertex in $V_2 \cap Q$ is connected to p. We will color the vertices of H with $|V_1|$ colors in the following way:

1. Color the vertices of V_1 with $|V_1|$ colors.
2. Color the vertices of $V_2 \cap Q$ with the color of p.
3. Color the vertices of type $v^2_{q_1,q_2,i}$ with the color of p.
4. Color the vertices of types $v^1_{q_1,q_2,i}, v^3_{q_1,q_2,i}$ with any color different from p.

If $|V_1| > 1$ we used $|V_1|$ colors. If $|V_1| = 1$ or $|V_1| = 0$ we used 2 colors so $\chi(H) = \omega(H)$ if the graph has edges. A graph with no edges is trivially perfect with a clique size 1 and coloring number 1 on any non empty induced subgraph. Therefore, for every induced subgraph H of $P(G)$, $\chi(H) = \omega(H)$ and by definition $P(G)$ is perfect. ∎

Lemma 2.11 *There is a perfect supergraph of $P(G)$, $Q(G) = (N, E' \cup F)$ such that $|F| \leq k$ iff there is a chain k-completion set of G. Moreover, every perfect k-completion set of $P(G)$ is a chain k-completion set of G.*

Proof. ($\Longleftarrow$) The previous lemma shows that $Q(G)$ is a perfect graph if adding F to G forms a chain graph.

($\Longrightarrow$) Suppose F is a perfect k-completion set for $P(G)$. Note first that $F \cap (Q \times Q) = \emptyset$ since if $(q_1, q_2) \in F$, $q_1, q_2 \in Q$, then q_1 and q_2 are incident on $k+1$ 5-cycles $\{q_1, v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}, q_2\}$ for $1 \leq i \leq k+1$, that cannot all be "destroyed" by the other edges in F. Let $F' = F \cap (P \times Q)$. Assume the contrary that $\hat{G} =$

$(P, Q, E \cup F')$ is not a chain graph. Therefore $\hat{G}$ contains a pair of independent edges: $(p_1, q_1), (p_2, q_2)$ such that $p_1, p_2 \in P$ and $q_1, q_2 \in Q$. But this means that the graph $Q(G)$ contains an induced C_7: There are $k+1$ induced C_7 subgraphs in $P(G) \cup F'$: $(\{(p_1, q_1, v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}, q_2, p_2) | 1 \leq i \leq k+1\})$.

Since $|F| \leq k$, the only way to "destroy" all the $k+1$ 7-cycles by adding edges from F, is by the edges $(p_1, q_2), (p_2, q_1)$ or (q_1, q_2). The first two edges are disallowed by the independence assumption and the last are ruled out by the first note above. ■

This concludes the proof that Perfect Completion is NP-hard. ■

Note that since the complexity of perfect graph recognition is open this proof does not imply NP-completeness.

2.2.2 Perfect Deletion

Theorem 2.12 *Perfect Deletion is NP-Hard.*

Proof. Reduction from Perfect Completion: Given a graph G and a constant k we form its complement $\bar{G}$ and use the same constant k.
By the perfect graph theorem, H is a perfect k-completion set of G iff $\bar{H}$ is a perfect k-deletion set of $\bar{G}$. ■

2.2.3 Perfect Editing

Theorem 2.13 *Perfect Editing is NP-Hard.*

Proof. Reduction from Chain Completion. The reduction is very similar to the reduction for Perfect Completion.
Given a bipartite graph $G = (P, Q, E)$ and a constant k we use the same constant k

and form a new graph $P(G) = (N, \hat{E})$, $N = P \cup Q \cup C \cup D$ where

$$C = \{v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i} | q_1, q_2 \in Q, q_1 \neq q_2, 1 \leq i \leq k+1\}$$

and

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

$\hat{E} = E \cup E_1 \cup E_2 \cup E_3$ where

$$E_1 = \{(u, v) | u, v \in P\}$$

and

$$E_2 = \{(q_1, v^1_{q_1,q_2,i}), (v^1_{q_1,q_2,i}, v^2_{q_1,q_2,i})(v^2_{q_1,q_2,i}, v^3_{q_1,q_2,i}), (v^3_{q_1,q_2,i}, q_2) | q_1, q_2 \in Q, 1 \leq i \leq k+1\}$$

$$E_3 = \{(p, w^1_{p,q,i}), (q, w^1_{p,q,i}), (p, w^2_{p,q,i})(w^2_{q_1,q_2,i}, w^3_{q_1,q_2,i}), (w^3_{q_1,q_2,i}, q) | (p, q) \in E_1 \cup E, 1 \leq i \leq k+1\}$$

In words: P is made a clique, any two Q vertices are connected by $k+1$ new disjoint 4-paths, and any edge with both ends in $P \cup Q$ is made a chord in $k+1$ new independent 5-cycles. For an example of the graph generated by the reduction see Figure 2.3.

Lemma 2.14 *If G is a chain graph then $P(G)$ is a perfect graph.*

Proof. The proof is along the lines of the proof Lemma 2.10 for Perfect Completion.
■

Lemma 2.15 *Every perfect k-editing set F of $P(G)$ satisfies $F \cap E = \emptyset$. Moreover, $F \cap (P \times Q)$ is a chain k-completion set for G.*

Proof. Let F be a perfect k-editing set for $P(G)$ and let the modified graph be $\hat{P}(G)$. Since each edge in $E \cup E_1$ is protected by $k+1$ C_5-s, the removal of such an edge is impossible. In particular, $F \cap E = \emptyset$. Note also that adding an edge to $Q \times Q$ is also impossible, since it would form $k+1$ C_5-s (involving Q and C vertices). Let $F' = F \cap (P \times Q)$. By the above arguments $F' \subseteq (P \times Q) \setminus E$. Suppose $(P, Q, E \cup F')$ is not a chain graph. Then it contains independent edges (p_1, q_1) and (p_2, q_2). But then p_1, q_1, p_2, q_2 are part of $k+1$ C_7-s which could not be destroyed in $\hat{P}(G)$, contradicting the assumption that $\hat{P}(G)$ is perfect. ■

In conclusion, there is a perfect k-editing of $P(G)$ iff there is a chain k-completion of G. ■

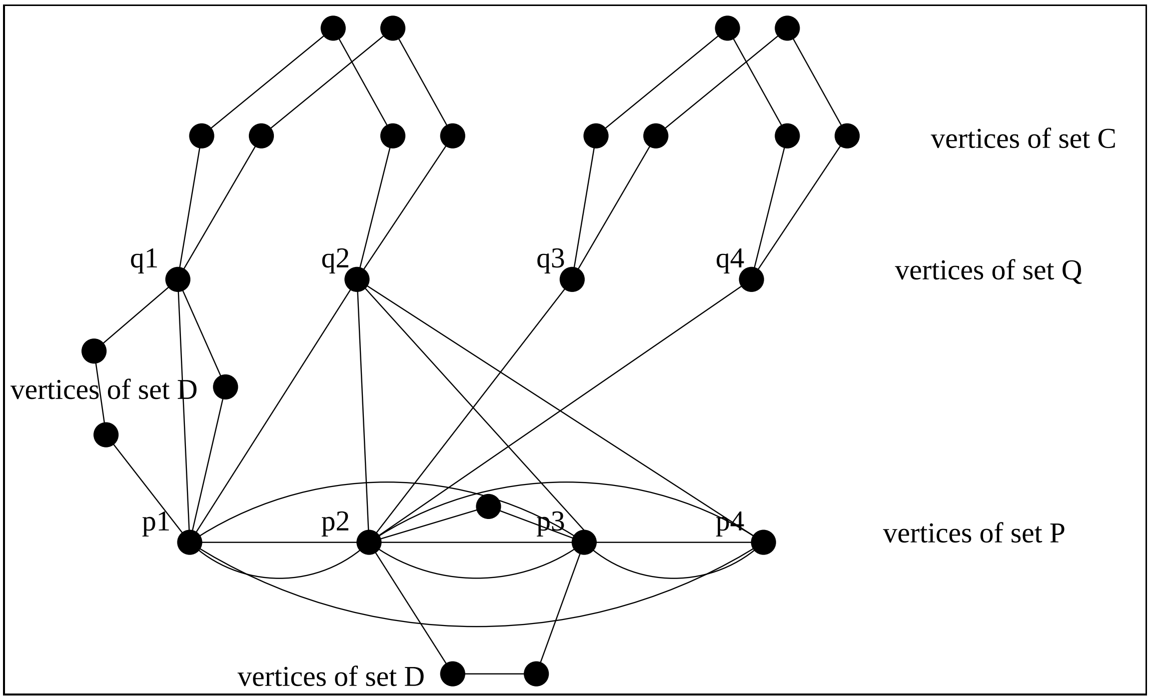

Figure 2.3: A part of the graph $P(G)$ in the proof of Theorem 2.13

The proofs for the perfect modification problems also apply for the Berge graph modification problems, therefore we can conclude:

Theorem 2.16 *Berge Graph Deletion, Completion and Editing are NP-hard.*

2.3 Comparability Editing

In this chapter we show that Comparability Editing is NP-complete, and prove also a hardness approximation result for it.

A *cut* in a graph $G = (V, E)$ is a partition (V_1, V_2) of V, *i.e.,* $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$. The *weight* of a cut (V_1, V_2) is the number of edges $(v, w) \in E$ so that $v \in V_1$ and $w \in V_2$. The *Max-Cut* problem is to decide, given a graph G and an integer k whether there is a cut whose weight is at least k. For an example of a graph with 4 edges and a cut of size 3 see Figure 2.4. The *Max-Cut* problem is NP-complete [15].

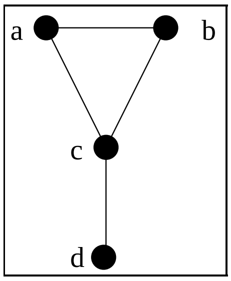

Figure 2.4: Max-Cut: In this graph the cut $(\{c\}, \{a, b, d\})$ has weight 3.

Theorem 2.17 *Comparability Editing is NP-complete.*

Proof. Reduction from Max-Cut. Given a graph $G = (V, E)$ and a constant k we use the constant $k_2 = |E| - k$ and form a new graph $C(G) = (N, \hat{E})$ where $N = V \cup E \cup W \cup Q$ and

$$W = \{w_i^v | v \in V, 1 \leq i \leq 4k_2 + 1\}$$

$$Q = \{e_{p,q}^1 | (p, q) \in E\}$$

and $\hat{E} = E_1 \cup E_2$ where

$$E_1 = \{(v, w_i^v) | v \in V, w_i^v \in W, 1 \leq i \leq 4k_2 + 1\}$$

$$E_2 = \{(v, e^1_{v,w}), (e^1_{v,w}, e_{v,w}), (e_{v,w}, w) | v, w \in V, (v,w) \in E\}$$

(for each $(v,w) \in E$ the choice of which one of w, v to connect to $e^1_{v,w}$ is arbitrary). Here we used e_{ab} to denote the vertex in $C(G)$ corresponding to the edge (a,b) in G. In other words, we attach to each original vertex in G $4k_2+1$ private neighbors, and replace each original edge by a path of length 3. For an example of a graph obtained by the reduction see Figure 2.5.

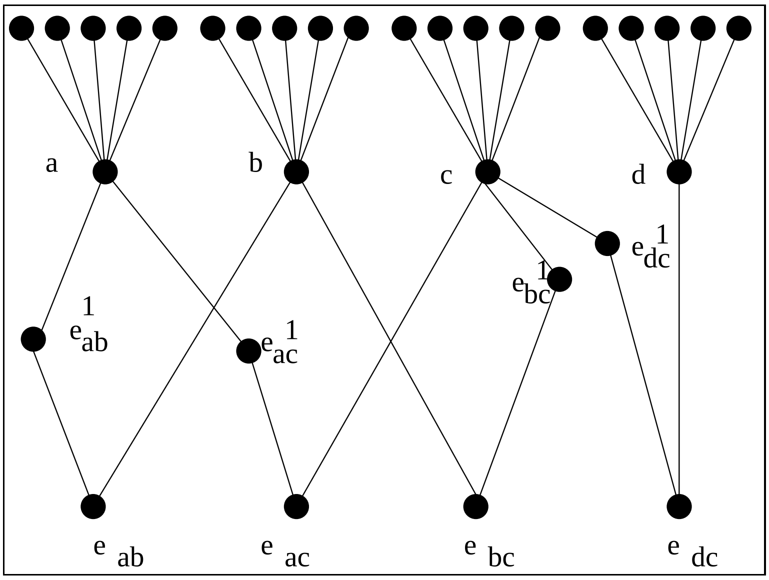

Figure 2.5: The graph C(G) built by the reduction applied to the graph of Figure 2.4 with $k = 3$.

We shall prove that there is a comparability k_2-editing $H = (N, R)$ of $C(G)$ iff there is a cut of weight k in G.

($\Longrightarrow$) Suppose (V_0, V_1) is a cut of weight k in G. For each non-cut edge

$$e = (v,w) \in ((V_0 \times V_0) \cup (V_1 \times V_1)) \cap E$$

we remove one edge of the corresponding path in $C(G)$: For the edge $e = (v,w)$ we remove the edge $(e_{v,w}, e^1_{v,w})$. We removed a total of $|E| - k$ edges from $C(G)$. Let the remaining graph be $Co(G) = (N, \hat{E}')$. We will build a transitive orientation of that graph, which implies that this graph is a comparability graph: Orient each arc incident on $v \in V_0$ (resp., $v \in V_1$) out of v (resp., into v). Orient the remaining edges along the paths to respect transitivity. The only orientation conflicts may arise if a path is intact (no edge was deleted in it). But then since the vertices at the ends of the path belong to different parts, the orientations along that path are transitive. For an example see Figure 2.6.

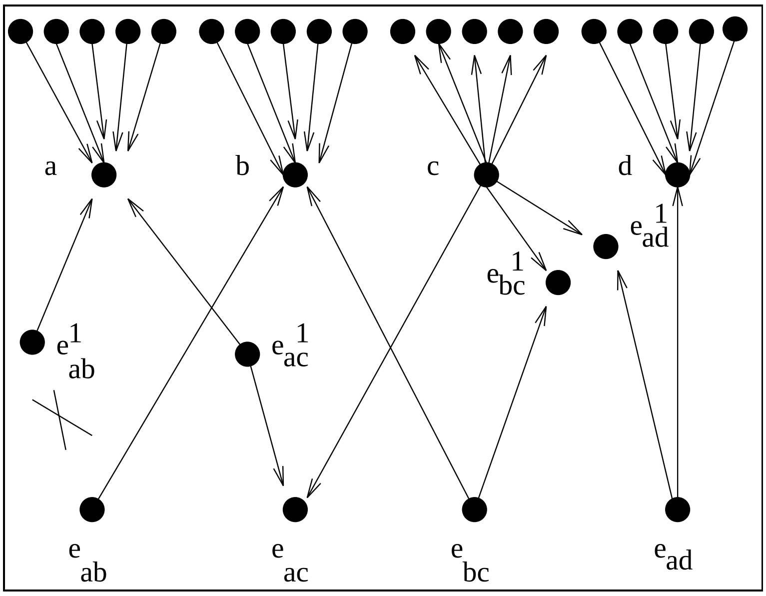

Figure 2.6: Orientation of the graph after removal of one edge

($\Longleftarrow$) Suppose $H = (N, R)$ is a comparability graph obtained by k_2-editing of $C(G)$. Let F be a transitive orientation of H.

Lemma 2.18 *For a vertex $v \in V$ either all arcs in F are incoming into v or they are all outgoing from v.*

Proof. Suppose H was formed by deleting l_1 edges and adding l_2 edges in $C(G)$. By assumption, $l_1 + l_2 \leq k_2$.

Let $W_1 = (w_v^i \in W | vw_v^i \in F)$, $W_2 = (w_v^i \in W | w_v^i, v \in F)$. Assume that $W_1 \neq \emptyset$ and $W_2 \neq \emptyset$. Since there are $4k_2+1$ of the type (v, w_v^i) in $C(G)$, and we deleted at most l_1 edges, either $|W_1| > \frac{4k_2+1-l_1}{2}$ or $|W_2| > \frac{4k_2+1-l_1}{2}$. W.l.o.g assume $|W_2| > \frac{4k_2+1-l_1}{2}$, and let $u \in W_1$. Since $uv \in F$ and for all $w \in W_2$ $vw \in F$, by transitivity of the orientation we added at least $|\frac{4k_2+1-l_1}{2}|$ edges to H. But $l_1 + |\frac{4k_2+1-l_1}{2}| > 2k_2$, a contradiction. ■

Define now a partition (V_0, V_1) of V by letting V_0 contain all the vertices incident on incoming arcs in F. We shall prove that there are at least k edges in the cut (V_0, V_1): For every edge $(v, w) \in E$ let us look at the set of three edges in the corresponding path

$$\{(v, e^1_{v,w}), (e^1_{v,w}, e_{v,w}), (e_{v,w}, w)\}$$

There are $|E|$ such pairwise disjoint edge sets. Since we modified at most $|E| - k$ edges from $\hat{E}$, there are at least k such paths that remained untouched, *i.e.,* with no change to the graph induced by the path of vertices. For each such path the edge (v, w) must in the cut (V_0, V_1): Otherwise, v and w are in the same side of the cut, which would form a contradiction to transitivity. ■

Note that the same reduction above with a very similar proof shows that Comparability Completion is NP-hard, a fact that was already shown in [21]. The same reduction also shows that Comparability Deletion is NP-hard, which was already shown by Yannakakis [41].

We now show that the problem is also hard for approximation:

Lemma 2.19 *If Comparability Editing can be approximated in polynomial time with ratio $1 + \theta$ ($\theta < 1$) then MAX-CUT can be approximated within a factor of $1 - \theta$.*

Proof. Suppose there is a polynomial approximation algorithm A for Comparability Editing that guarantees an approximation ratio of $1 + \theta$ for some $\theta < 1$. To approximate Max-Cut on a graph $G = (V, E)$, apply to G the reduction in Theorem 2.17

with k set to zero and obtain the graph $C(G)$. Apply algorithm A to $C(G)$ and obtain an edited graph $C(\hat{G})$ that has a transitive orientation F.

We argue first that for every $v \in V(G)$ either all arcs in F are oriented into v, or all are oriented out of v. For proof, observe that a maximum cut in G has value $k^* \geq \frac{|E|}{2}$, since every graph has a cut with at least half its edges. A minimum editing in $C(G)$ has size k^* at most $|E|$ (delete one edge in each 3-path) so the approximation algorithm gives a solution with value at most $2|E|$. The same argument as in Lemma 2.18 proves the claim.

Define a partition (V_0, V_1) of V by letting V_0 contain all vertices with incoming arcs according to F. The number of deleted edges in $C(G)$ is at most $(1+\theta)k^*$, so at least $|E| - (1+\theta)k^*$ 3-paths corresponding to E edges remain unchanged, and the corresponding cut in G has cardinality at least $|E| - (1+\theta)k^*$. By the proof of Theorem 2.17, the value of the maximum cut in G, k^{opt} satisfies $k^{opt} = |E| - k^*$. This implies that the approximation ratio of our solution is at least $\frac{|E|-(1+\theta)k^{opt}}{k^{opt}} = \frac{|E|}{k^{opt}} - (1+\theta) \geq 2 - (1+\theta) = 1 - \theta$, where the inequality follows since $k^{opt} \geq \frac{|E|}{2}$. ■

Since approximation of Max-Cut to within a factor of $\frac{17}{18}$ is NP-hard [40, 24], we conclude:

Theorem 2.20 *It is NP-hard to approximate Comparability Editing to within a factor of $\frac{19}{18}$.*

As the same reduction above applies to Comparability Completion and Deletion we also obtain the following result.

Theorem 2.21 *It is NP-hard to approximate Comparability Completion and Comparability Deletion to within a factor of $\frac{19}{18}$.*

2.4 Cluster Deletion

The Cluster Completion problem is trivially polynomial: we simply have to transform every connected component into a clique.

Theorem 2.22 *Cluster Deletion is NP-complete.*

Proof. Reduction from Clique. In the Clique problem, given an input $G = (V, E)$ and a constant k, the question is whether G contains an induced k-clique. This problem is NP-hard [28]. Let $|V| = n$. We build a new graph $H = (V \cup W, E \cup F)$, $|W| = n^2$ and $F = \{(v, w) | v \in V \cup W, w \in W\}$. In other words, we add to G an n^2-clique and fully connect it with V. We set the constant to $k_2 = n^2(n-k) + \binom{n}{2}$.

($\Longrightarrow$) Let K_k be a clique of size k in G. Deleting the edges $D = \{(v, w) \in E \cup F | v \in V \setminus K_k, w \in V \cup W\}$, we get a cluster graph, and $|D| \leq n^2(n-k) + \binom{n}{2} - \binom{k}{2} < k_2$.

($\Longleftarrow$) Let $\hat{H}$ be a cluster graph obtained by deleting k_1 edges from H, where $k_1 \leq k_2$. Let R be the subgraph of $\hat{H}$ induced by W. Let X be a maximum clique in R. If $|X| \leq n^2 - n$ then at least $n(n^2 - n) > k_2$ edges were deleted from the induced subgraph H_W. So $|X| > n^2 - n$. Let $|X| = n^2 - t$, $t < n$. If all the cliques in G are of size $k-1$ or less, then at least $n - k + 1$ vertices do not belong to same clique as X in $\hat{H}$. Therefore we deleted at least

$$(n - k + 1)(n^2 - t) + t(n^2 - t) \geq (n - k)n^2 + n^2 > k_2$$

edges, a contradiction. ■

2.5 Cograph Deletion and Completion

Theorem 2.23 *Cograph Deletion is NP-complete.*

Proof. By reduction from Chain Deletion. Let $< G = (P, Q, E), k >$ be an instance of Chain Deletion. Build the following instance $< G' = (V, E'), k >$ of Cograph Deletion: Define $V = P \cup Q \cup W$, where $W = \{w_0, ..., w_k\}$. Define $E' = E \cup \{(p_1, p_2) | p_1, p_2 \in (P \cup W)\}$.

$\Leftarrow$ Let F be a solution to the chain instance. If $\hat{G} = (V, E' \cup F)$ is not a cograph $\hat{G}$ contains an induced P_4, (v_1, v_2, v_3, v_4). Since $P \cup W$ is a clique and Q is an independent set, $v_1, v_4 \in Q$ and $v_2, v_3 \in P \cup W$. This means that (v_1, v_2) and (v_3, v_4) are two independent edges, a contradiction.

$\Rightarrow$ If F is a solution to the cograph instance and $F \cap E$ is not a solution to the chain instance, then there exist a pair of independent edges e_1, e_2 in $(P, Q, E \setminus F)$. Let $e_1 = (p_1, q_1)$, $e_2 = (p_2, q_2)$, $p_1, p_2 \in P$, $q_1, q_2 \in Q$. If we did not remove the edge (p_1, p_2) from G', (q_1, p_1, p_2, q_2) is an induced P_4. Otherwise, for some $0 \leq i \leq k$ we did not remove the edges (w_i, p_1), (w_i, p_2) (since we removed at most k edges), and thus $(q_1, p_1, w_2, p_2, q_2)$ is an induced P_5 in $(V, E' \setminus F)$, a contradiction.

■

The complement of a cograph is a cograph, therefore we conclude:

Theorem 2.24 *Cograph Completion is NP-complete.*

2.6 Split Graphs Deletion and Completion

Theorem 2.25 *Split Deletion in NP-complete.*

Proof. 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 $W = \{w_1, \dots, w_{n^2+1}\}$, and define $E' = E \cup (V \times W)$.

$\Leftarrow$ 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 $n^2(n-k) + \binom{n-k}{2} < n^2(n-k+1)$.

$\Rightarrow$ Assume 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' ($|W \setminus K| \geq n^2$ since W is an independent set and K is a clique, $|V \setminus K| \geq n-(k-1)$ since $|V \cap K| \leq k-1$), a contradiction.

■

The complement of a split graph is a split graph, therefore, we conclude:

Theorem 2.26 *Split Completion is NP-complete.*

Chapter 3

Approximation Algorithms

3.1 Modification problems on bounded degree graphs for families characterized by a finite set of forbidden induced subgraphs

We present below a polynomial time constant factor approximation algorithm for the Deletion and Editing problems on bounded degree graphs. The result applies to any hereditary family that can be characterized by a finite set of forbidden induced subgraphs. These include among others, cographs, trivially perfect graphs, split graphs, threshold graphs, Meyniel graphs, claw-free graphs, superfragile graphs, maxibrittle and superbrittle graphs, cf. [9]. (Note that for some of these families the complexity of Editing and Deletion is open, and they may, in fact, be polynomial). An analogous result for vertex deletion problems was given by Lund and Yannakakis [32].

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 at most d.

3.1.1 Edge Deletion approximation algorithm

We distinguish three cases:

Case 1: No forbidden subgraph contains an isolated vertex. We use $V(G)$ to denote the vertex set of the graph G. The approximation algorithm follows.

Algorithm $A(G, \mathcal{F})$

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 a forbidden induced subgraph with t vertices can be found in $O(n^t)$ time.

Theorem 3.1 *The algorithm achieves an approximation ratio 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, upon termination of the algorithm the remaining subgraph has property Π.

Approximation ratio: Let F be an optimum solution set 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. ■

Case 2: The set $\mathcal{F}$ contains graphs with isolated vertices, but no forbidden subgraph is an independent set. In case $n \leq 2dt$, the problem is solved exactly by exhaustive search. Otherwise, we define a new set of forbidden graphs $\mathcal{F}'$ by removing the isolated vertices from the original forbidden graphs. Algorithm A is then applied to the input $(G, \mathcal{F}')$.

Let F be a forbidden graph with some isolated vertices. Let F' be the graph after removing the isolated vertices.

Lemma 3.2 *Let G be a graph with $n > 2dt$ vertices and bounded degree d. Either G contains an induced copy of F or it does not contain an induced copy of F'*

Proof. Assume the contrary. Let H be an induced copy of F'. Let S denote the set of vertices not adjacent to any vertex in H. Since $|H| < t-1$, $|S| > n-(t-1)d > dt$. Since the graph induced on S also has degree bounded by d, it contains an independent set S' of size $t-1$. Hence, $S' \cup H$ contains an induced copy of F. ■

Theorem 3.3 *The algorithm achieves an approximation ratio of td.*

Proof. Let $\hat{G}$ be a solution to the edge deletion problem, where $|V(\hat{G})| > 2dt$. By the previous lemma $\hat{G}$ cannot contain a forbidden subgraph in $\mathcal{F}'$, therefore by Theorem 3.1 the algorithm achieves an approximation ratio of td. ■

Case 3: The set $\mathcal{F}$ contain a graph which is an independent set. If $n \leq (d+1)t$, the problem is solved exactly by exhaustive search. Otherwise, there is no solution for the deletion problem, since a graph with $(d+1)t$ vertices and bounded degree d contains an induced independent set of size t.

3.1.2 Edge Editing approximation algorithm

Again, we distinguish three cases:

Case 1: No forbidden subgraph contains an isolated vertex. In this case we use Algorithm $A(G, \mathcal{F})$.

Theorem 3.4 *The algorithm achieves an approximation ratio of td.*

The proof is along the lines of the Proof 3.1

Case 2: The set $\mathcal{F}$ contains graphs with isolated vertices, but no forbidden subgraph is an independent set. We define a new set of forbidden graphs $\mathcal{F}'$ by removing the isolated vertices from the original forbidden graphs. The approximation algorithm solves the problem exactly by exhaustive search if $n \leq 4dt$, and otherwise applies Algorithm $A(G, \mathcal{F}')$. In the analysis we concentrate on the situation when $n > 4dt$.

Lemma 3.5 *If the minimum editing graph contains a copy of $F' \in \mathcal{F}'$, the minimum editing size is at least $\frac{n-2dt}{2}$.*

Proof. Let G be the original graph and let G' be a minimum editing graph. Suppose G' contains an induced copy H of F'. Let $N_G(H) = \{v \in G \setminus H | (u,v) \in E$ for some $v \in H\}$. Then $|H| \leq t-1$ and $|N_G(H)| \leq (t-1)(d-1)$. Let $S = V \setminus (H \cup N_G(H))$. Then $|S| \geq n-(t-1)-(t-1)(d-1) = n-(t-1)d > n-td$. In G'_S at least $|S|-td$ vertices have new edges incident on them, or else G' contains a copy of F. (G' contains an independent set of size $t-1$ since there are at least td vertices in S with no added edges, and a graph with td vertices and degree bounded by d must have an independent set of size $t-1$). This implies that at least $\frac{n-2td}{2}$ edges were added. ■

Theorem 3.6 *The algorithm achieves an approximation ratio of td.*

Proof. Let G be the original graph, and let G' be a minimum editing graph. If G' contains no copy of $F' \in \mathcal{F}'$, by Theorem 3.4 the approximation algorithm achieves an approximation ratio of td. Otherwise, by the previous lemma at least $\frac{n-2td}{2}$ edges must be edited in G'. Since Algorithm A only deletes edges, the number of edges it deleted is no more than all the edges in G, which is at most $\frac{nd}{2}$. Hence the approximation ratio is at most $\frac{nd/2}{(n-2dt)/2)} < \frac{nd}{n/2} = \frac{d}{2} < td$. ■

Case 3: The set $\mathcal{F}$ contains a graph which is an independent set. We first claim that in that case $\mathcal{F}$ cannot contain also another F' that is a clique because by Ramsey's Theorem cf. [8] a graph with no independent set of size l and no clique of size k has at most a constant number of vertices. For a graph smaller than that number, the problem is solved exactly, otherwise, the editing problem approximation solution is to add all the edges, *i.e.,* to turn G into a clique.

Lemma 3.7 *A graph with no independent set of size l+1 has at least* $l\binom{\frac{n}{l}}{2}$ *edges.*

Proof. Suppose the largest independent set in any induced subgraph of $G = (V, E)$ is l. Consider a largest independent set S of G. By the maximality of S each of the vertices of $V \setminus S$ is connected to a vertex is S. Therefore there are at least $n - l$ edges between S and $V \setminus S$. The graph $G_{V \setminus S}$ is an induced subgraph of G, therefore, the same argument applies. By induction, the number of edges in G is at least $\sum_{i=1}^{\frac{n}{l}-1} li = l\binom{\frac{n}{l}}{2}$. ■

An example of a graph with $l\binom{\frac{n}{l}}{2}$ edges and no independent set of size $l+1$ is $l * K_{\frac{n}{l}}$.

Theorem 3.8 *The algorithm achieves an approximation ratio of t.*

Proof. Let G be the original graph. Let G' be a solution to the minimum editing problem. By the previous lemma G' contains at least $t\binom{\frac{n}{t}}{2}$ edges, *i.e.,* at least $t\binom{\frac{n}{t}}{2} - \frac{nd}{2}$ edges were added. Since the approximation algorithm adds at most $\binom{n}{2}$ edges, it achieves a ratio of t. The algorithm generates a feasible solution since $\mathcal{F}$ cannot contain a clique if n is large enough. ■

We can now summarize the results of the two previous subsections:

Theorem 3.9 *For the Deletion and Editing problems with respect to any graph property that is characterized by a finite set of forbidden induced graphs, there exist a polynomial time constant factor approximation algorithm on bounded degree graphs.*

3.2 Cluster Deletion and Editing

In this chapter we present approximation algorithms for Cluster Deletion and Cluster Editing. The family $\mathcal{H}$ of cluster graphs is characterized by the forbidden induced subgraph P_3, *i.e.,* a path of length two. Such a path can be found in linear time using breadth first search.

3.2.1 Cluster Editing

The algorithm for the editing problem is described below. The input is a graph $G = (V, E)$.

0. $A = \emptyset$

1. While there exist $W \subseteq V$ that induces a P_3 set $V \leftarrow V \setminus W, A \leftarrow A \cup W$

2. Denote by $\hat{V}$ the set V after Step 1 is complete. ($G_{\hat{V}}$ is a cluster graph). For each $x \in A$ compute a minimum editing set E_x on $G_{\hat{V} \cup \{x\}}$, and determine the clique in $G_{\hat{V}}$ associated with x by this editing.

3. Perform the editing for each $x \in A$ independently.

4. For each $x, y \in A$ if x, y are adjacent and associated with different cliques remove (x, y). If x, y are non-adjacent and associated with the same clique add (x, y).

Claim 3.10 *For each $x \in A$ a minimum edit set E_x can be found in polynomial time in Step 2. Moreover, there is such a set that includes only edges incident on x.*

Proof. Let $C_1, .., C_l$ be the cliques in an optimal editing of $G_{\hat{V} \cup \{x\}}$. W.l.o.g assume that $x \in C_1$. Let $C_1' = C_1 \setminus \{x\}$. If C_1' contains vertices of two or more cliques in $G_{\hat{V}}$, then $C_1' = A_1 \cup A_2$, such that the vertices of A_1 belong to one clique and the vertices of A_2 belong to other cliques. Removing the edges we added between the vertices of A_1 and A_2, and removing the edges between x and A_2 we get a cluster graph with no more edits (we did at most $|A_2|$ new edits and canceled at least $|A_1||A_2|$ added edges). Therefore we can assume that C_1' contains vertices of only one clique in $G_{\hat{V}}$. In the same way we can show that C_1' can contain all the vertices of that clique, *i.e.,* we can assume C_1' is a clique in $G_{\hat{V}}$. Since we can assume that C_1' is a clique in $G_{\hat{V}}$ we can also assume that $C_2, ..., C_l$ are cliques in $G_{\hat{V} \cup \{x\}}$ because changing them will only increase the number of edits.

Computing a minimum editing set E_x can be done by calculating for each clique in $G_{\hat{V}}$ the number of edits needed to be done if we add x to it, *i.e.,* for each clique we have to add the missing edges between x and the clique and delete the edges between x and other cliques. This can be done in linear time, for all $x \in A$. ■

For $x \in A$, let $k_x = |E_x|$ as defined in Step 2 of the algorithm. Define $k_2 = Max\{k_x | x \in A\}$. Let t^* be the size of a minimum editing set for G.

Claim 3.11 $t^* \geq Max\{\frac{|A|}{3}, k_2\}$.

Proof. Let $y = argmax\{k_x | x \in A\}$. Since the subgraph induced by $G_{\hat{V} \cup \{y\}}$ must be a cluster graph at least k_2 edit operations are needed. Since on any induced P_3 in A an edit operation must be performed, it follows that $t^* \geq \frac{|A|}{3}$. ■

Claim 3.12 *The algorithm performs at most $\binom{|A|}{2} + |A|k_2$ edge edit operations.*

Proof. Step 3 requires at most k_2 edits per vertex in A. Step 4 requires at most $\binom{|A|}{2}$ edits. ■

Theorem 3.13 *The algorithm finds a cluster graph and guarantees an approximation ratio of* $7.5t^*$.

Proof. The algorithm does at most $\binom{|A|}{2} + |A|k_2$ edits. Since $t^* \geq Max\{\frac{|A|}{3}, k_2\}$ we obtain the ratio: $\frac{\binom{|A|}{2}+|A|k_2}{t^*} = \frac{\binom{|A|}{2}}{t^*} + \frac{|A|k_2}{t^*} \leq 3\frac{\binom{|A|}{2}}{|A|} + |A| \leq 2.5|A| \leq 7.5t^*$. ■

3.2.2 Cluster Deletion

The algorithm for the deletion problem is given below. The input is a graph $G = (V, E)$.

0. $A = \emptyset$
1. While there exist $W \subseteq V$ that induces a P_3 set $V \leftarrow V \setminus W, A \leftarrow A \cup W$
2. Denote by $\hat{V}$ the set V after Step 1 is complete. For each $x \in A$ compute a minimum deletion set E_x on $G_{\hat{V} \cup \{x\}}$, and determine the clique C_x in $G_{\hat{V}}$ associated with x by this deletion.
3. Perform the deletion for each $x \in A$ independently.
4. For each $x, y \in A$ if x, y are adjacent and associated with different cliques remove (x, y). If x, y are non-adjacent and associated with the same clique remove all edges incident on either x or y.

Claim 3.14 *For each $x \in A$ a minimum deletion set E_x can be found in polynomial time in Step 2. Moreover, there is such a set that includes only edges incident on x.*

The proof is along the lines of the proof of Claim 3.10.

For $x \in A$, let $k_x = |E_x|$ as defined in step 2 of the algorithm. Define $k_2 = Max\{k_x | x \in A\}$. A clique C_x is called *split* if there are $x, y \in A, (x, y) \notin E$ and $C_x = C_y$. In that case x and y are called *splitting vertices* for C_x. Let $k_3 = Max\{|C_x| | C_x$ is a split clique $\}$. Let t^* be the minimum number of edge deletion operations needed on G.

Claim 3.15 $t^* \geq Max\{\frac{|A|}{3}, k_2, k_3\}$.

Proof. Let C_x be a maximum split clique, and let x, y be splitting vertices for C_x. Since the induced subgraph $G_{C_x \cup \{x,y\}}$ must be transformed to a cluster graph, at least k_3 edges must be deleted. The rest of the proof is identical to Claim 3.11. ∎

Theorem 3.16 *The algorithm finds a cluster graph with guaranteed approximation ratio of* $10.5t^*$.

Proof. In Step 3, at most k_2 deletions are needed per vertex in A. Step 4 requires at most $\binom{|A|}{2}$ deletions between vertices of A and at most $|A|k_3$ more deletions for splitting vertices. Therefore, the algorithm does at most $\binom{|A|}{2} + |A|(k_2 + k_3)$ deletions. Since $t^* \geq Max\{\frac{|A|}{3}, k_2, k_3\}$ the ratio is at most $7.5t^* + \frac{k_3}{t^*}|A| < 7.5t^* + 3t^* < 10.5t^*$. ∎

Chapter 4

Polynomial Results On Bounded Degree Graphs

We present below polynomial algorithms for Chain Editing and Threshold Editing on bounded degree graphs.

Lemma 4.1 *Let Ψ be a hereditary graph property, such that if $G = (V, E)$ satisfies Ψ $G_{V \setminus \{v\}} \cup \{v\}$ satisfies Ψ (i.e., the property remains satisfied if we remove all the edges incident on a vertex v). Any optimal solution of Ψ-Editing on a graph with bounded degree d is a graph with degree bounded by $2d$.*

Proof. Let $G = (V, E)$ be an instance to the Ψ-Editing problem. Assume that a minimum Ψ-Editing of G contains a vertex w with degree at least $2d+1$. This means we added at least $d+1$ edges incident on w, but if instead we removed all the edges incident on w we would have performed less edits and the resulting graph has the property Ψ, a contradiction. ■

Lemma 4.2 *A chain graph $G = (P, Q, E)$ with bounded degree d contains at most $2d$ vertices with degree at least one.*

Proof. Let W be the set of vertices with degree at least one. Let $P' = P \cap W$ and let $Q' = Q \cap W$. There is a vertex $q \in Q$, $q \in \{\cap Adj(p) | p \in P'\}$ such that q has degree $|P'|$. Similarly there is a vertex $p \in P'$ with degree $|Q'|$. If $|P' \cup Q| > 2d$, either $|P'| > d$ or $|Q'| > d$, so there is a vertex with degree at least $d+1$, a contradiction. ■

Theorem 4.3 *Chain Editing on bounded degree graphs is polynomial.*

Proof. Let $G = (V, E)$ be an instance of Chain Editing, and let $H = (V, E')$ be the graph obtained by a minimum chain editing of G. If G has degrees bounded by d, by Lemma 4.1, H has degree bounded by $2d$. By Lemma 4.2 there are at most $4d$ vertices with degree at most one in H. The polynomial algorithm will enumerate all sets of at most $4d$ vertices and for each such set it will solve exactly the Chain Editing problem. The value of each solution is the optimal value of the editing problem on S plus the number of edges not incident on S (at both endpoints). The best value obtained is the resulting solution. The algorithms runs in $O(n^{4d})$ time. ■

Theorem 4.4 *Threshold Editing on bounded degree graphs is polynomial.*

Proof. T is a threshold graph if $T = (K, I, E)$ such that K induces a clique, I induces an independent set, and the graph $(K, I, E \cap (K \times I))$ is a chain graph. Let $G = (V, E)$ be an instance of Threshold Editing , and let $H = (K, I, E')$ be the graph obtained by a minimum threshold editing of G. If G has degrees bounded by d, by Lemma 4.1 H has degree bounded by $2d$. H contains a vertex in K which is adjacent to all non-isolated vertices in $K \cup I$. Since its degree is at most $2d$, there are at most $2d + 1$ non-isolated vertices in H. The algorithm will enumerate the set K and the vertices with degree at least one in I. For each such set of at most $2d+1$ vertices the algorithm calculates the minimum threshold editing. The algorithm runs in $O(n^{2d+1})$ time. ■

Chapter 5

Combinatorial bounds on the size of minimum completion sets

A natural combinatorial question arises when dealing with completion problems: How large can the completion set size be compared to the original graph size? As usual, for a graph $G = (V, E)$ we denote $|V| = n, |E| = m$. For a family of graphs $\mathcal{F}$ define $\Phi_{\mathcal{F}}(m, n) = max_G min\{|\bar{E}| : G(V, E \cup \bar{E}) \in \mathcal{F}\}$. In this notation, our question is: How large can $\Phi_{\mathcal{F}}(m, n)$ be as a function of m and n?

5.1 A concrete example for Chordal Completion

Let $\mathcal{F}$ be the family of chordal graphs. We will build a graph $G = (V, E)$ where $m = \Theta(n^{\frac{3}{2}})$, and $\Phi_{\mathcal{F}}(m, n) = \Theta(n^2)$.

Let $G = (V, W, E)$ be the following bipartite graph: $V = (v_1, v_2, \ldots, v_n)$, $W = (w_1, w_2, \ldots, w_n)$ and $E = \{e_i = (v_i, w_i) | 1 \le i \le n\}$. In words, G is a perfect matching on a $2n$-vertex graph, or nK_2.

Lemma 5.1 *The smallest chain graph containing G has $\Theta(n^2)$ edges.*

Proof. Let $H = (V, W, \hat{E})$ be the smallest chain graph containing G (W.l.o.g we can assume that the bipartition of H is also (V, W).). Since H has the chain property, the vertices of V can be ordered v_1,v_2,..,v_n so that $Adj(v_1) \supseteq Adj(v_2) \supseteq ... \supseteq Adj(v_n)$. W.l.o.g vertex numbers v_1,v_2,..,v_n satisfy this property in the graph H. Since $(v_i, w_i) \in E$ we have $w_i \in Adj(v_i)$. Since $Adj(v_i) \supseteq Adj(v_j)$ if $j < i$, it follows that $Adj(v_j) \supseteq (w_{j+1}, \ldots, w_n)$. Therefore, $|Adj(v_i)| \geq n - i$. In fact, $|Adj(v_i)| = n - i$ is a minimal chain graph containing G and $|\hat{E}| = \sum_{i=1}^{n}(n-i) = \frac{n(n-1)}{2} = \Theta(n^2)$. ■

For a bipartite graph $G = (P, Q, E)$ we define a new graph $C(G) = (N, E')$, where $N = P \cup Q$, $E' = E \cup \{(u, v) | v \in P\} \cup \{(u, v) \mid u, v \in Q\}$ (*i.e.,* we replace the independent set on each part by a clique.) See Figure 5.1.

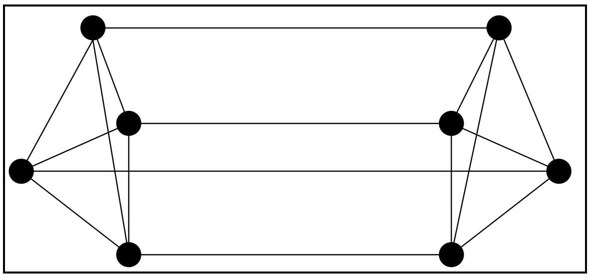

Figure 5.1: $C(G)$ for the graph $G = 4K_2$

We shall need the following lemma, due to Yannakakis:

Lemma 5.2 *[42] G is a chain graph iff $C(G)$ is triangulated.*

We are now ready to construct our concrete family of examples. Let $G = (V, E)$ be the following graph: $V = \cup_{i=1}^{n} W_i$ where

$$W_i = \{w_i^1, w_i^2, \ldots w_i^n\}$$

and

$$E = \{(w_i^j, w_i^k) | 1 \leq i, j, k \leq n\} \cup \{(w_j^i, w_k^i) | 1 \leq i, j, k \leq n\},$$

i.e., the graph

$$G_i = (W_i, E \cap (W_i \times W_i))$$

is a clique and the subgraph

$$G^{i,j} = (W_i \cup W_j, E \cap (W_i \times W_j))$$

is isomorphic to nK_2.

The graph $G = (V, E)$ has n^2 vertices and

$$|E| = n^3 + n\binom{n}{2} = \Theta(n^3).$$

For an example of the graph we build see Figure 5.2.

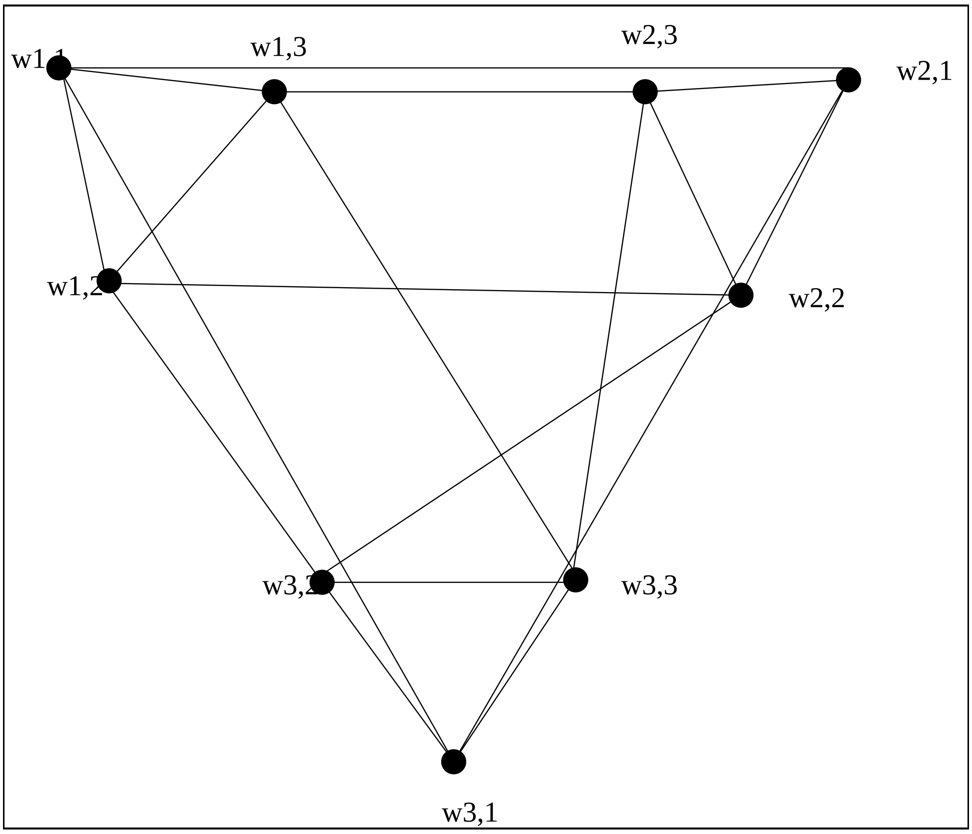

Figure 5.2: the graph we construct in 5.1 for n=3

Claim 5.3 *A smallest chordal completion set of G contains $\Theta(n^4)$ edges.*

Proof. Let $H = (V, \hat{E})$ be a smallest chordal graph containing G. Since chordality is hereditary, every subgraph of H is chordal. Let us look at the subgraph $H_{i,j}$ of H induced by $W_i \cup W_j$. $H_{i,j}$ is a chordal supergraph of $G^{i,j}$, therefore by Lemma 5.2 $H^{i,j} = (W_i \cup W_j, \hat{E} \cap (W_i \times W_j))$ is a chain supergraph of $G^{i,j}$. By Lemma 5.1 $\Theta(n^2)$ edges must have been added between W_i and W_j. Since there are $\binom{n}{2}$ different subgraphs $\{H_{i,j} | i \neq j\}$ and in each such subgraph different edges must be added, $\Theta(n^4)$ edges were added. ■

We note that Kaplan and Szegedi [27] have recently argued that for certain expander graphs $m = \Theta(n)$ and $\Phi_{\mathcal{F}}(m, n) = \Theta(n^2)$. In contrast, our example is elementary and does not rely on the deep expander theory.

5.2 Random graphs

In this chapter we study probabilistically the size of a minimum completion set in some random graphs. For background on random graphs see [3]. We start with the definition of a random graph:

Let n be a positive integer, and let $0 \leq p \leq 1$. A random graph $G(n,p)$ is a probability space over the set of graphs on the vertex set $\{1, \ldots, n\}$ determined by

$$Pr[(i,j) \in G] = p$$

independently for each pair (i,j).

We give below some basic bounds on large deviations, which will be used in this chapter. The results are based on the seminal paper of H. Chernoff [11].

Theorem 5.4 *Chernoff bounds [11]: Let $p \in [0,1]$, $X_1, \ldots, X_n$ be mutually independent random variables with $Pr[X_i = 1] = p$ and $Pr[X_i = 0] = 1 - p$, and define $X = X_1 + \ldots + X_n$. Then:*

1. $Pr[X - np > a] < e^{a - pnln(1+\frac{a}{pn}) - aln(1+\frac{a}{pn})}$.

2. If $p = \frac{1}{2}$ then $Pr[|X - \frac{n}{2}| > a] < 2e^{\frac{-2a^2}{n}}$

3. When $p = o(1)$, $a = O(pn)$, and $a > \frac{2pn}{3}$ we have a simple bound:

$$Pr[X - np > a] \leq Pr[X - np > \frac{2pn}{3}] < e^{\frac{-2pn}{27}}.$$

Theorem 5.5 *Chebychev Inequality [39]: For any positive* λ

$$Pr\left[|X - E[X]| \geq \lambda\sqrt{Var[X]}\right] \leq \frac{1}{\lambda^2}$$

An event A on a family of random graphs $G(n,p)$ happens *with high probability* if $\lim_{n\to\infty} Pr(A) = 1$.

5.2.1 The minimum chordal completion of a random graph

In this chapter we study the random graph $G(n, \frac{1}{\sqrt{n}})$. We shall show that with high probability G has $O(n^{1.5})$ edges but the smallest chordal graph that contains it has $\Omega(\frac{n^2}{log^2(n)})$ edges.

Claim 5.6 *With high probability the random graph* $G(n, \frac{1}{\sqrt{n}})$ *has* $O(n^{1.5})$ *edges.*

Proof. Let m denote the random variable of the number of edges in G. Define a random variable e_i for each edge i where $e_i = 1$ if $i \in E$ and $e_i = 0$ otherwise. Then $m = \sum_{i=1}^{\binom{n}{2}} e_i$. Since the variables are independent and $Pr(e_i = 1) = \frac{1}{\sqrt{n}}$, $E(m) = \binom{n}{2} * \frac{1}{\sqrt{n}}$. By Theorem 5.4 (3) (taking, say, $a = n^{1.5}$), $Pr[m > E(m) + n^{1.5}] = Pr[m > \frac{1}{\sqrt{n}}\binom{n}{2} + n^{1.5}] < e^{-\frac{2n^{1.5}}{27}}$. So with high probability $m = O(n^{1.5})$. ■

Claim 5.7 *Let* X *be the number of labeled 4-cycles in* G *(not necessarily induced). With high probability* $X = \Theta(n^2)$.

Proof. The number of possible 4-cycles in a graph is $\frac{3!}{2}\binom{n}{4}$ (we have to choose the four vertices of the cycle and order them). For every possible 4-cycles we will define an indicator X_i, $0 < i \leq \frac{3!}{2}\binom{n}{4}$ with $X_i = 1$ if that cycle is present in G and $X_i = 0$ otherwise. Define $X = \sum X_i$. By linearity of expectation we have

$$E[X] = \frac{3!}{2}\binom{n}{4}p^4$$

i.e., $E[X] = \Theta(n^2)$.

$$Var[X] = \sum Var[X_i] + \sum_{i \neq j} Cov[X_i, X_j]$$

since X_i is an indicator

$$Var[X_i] = p_i(1 - p_i) \leq p_i = E[X_i]$$

we have

$$Var[X] \leq E[X] + \sum_{i \neq j} Cov[X_i, X_j].$$

For indices i, j, we shall write $i \sim j$ if $i \neq j$ and the events X_i, X_j are not independent. If $i \sim j$

$$Cov[X_i, X_j] = E[X_iX_j] - E[X_i]E[X_j] \leq E[X_iX_j] = Pr[X_i \wedge X_j]$$

and therefore

$$Var[X] \leq E[X] + \sum_{i \sim j} Pr[X_i \wedge X_j]$$

In our case $i \sim j$ if X_i and X_j are different and have common edges. X_i and X_j may have one or two common edges. There are $O(n^6)$ sets with one common edge and the probability for each such set is p^7, and there are $O(n^5)$ sets with two common edges and the probability for each such set is p^6.
Therefore

$$Var[X] = O(n^2) + O(n^6p^7) + O(n^5p^6) = O(n^{2.5})$$

Using Chebychev Inequality (Theorem 5.5), with $\lambda = \sqrt{n}$

$$Pr\left[|X - E[X]| > \sqrt{n}\sqrt{Var[X]}\right] \leq \frac{1}{n}. \tag{5.1}$$

Hence, with probability larger than $1 - \frac{1}{n}$,

$$|X - E[X]| \leq \sqrt{nVar[X]}$$

Since $Var(X) = O(n^{2.5})$, for n sufficiently large $Var(X) < Cn^{2.5}$ for some constant C. Since $E(X) = \Theta(n^2)$, for n sufficiently large, $An^2 < E(X) < Bn^2$ for some constants A, B. Since w.h.p. $|X - E[X]| \leq \sqrt{nVar[X]} \leq \sqrt{nCn^{2.5}}$, it follows that for n sufficiently large $An^2 - \sqrt{C}n^{1.75} < X < Bn^2 + \sqrt{C}n^{1.75}$. Therefore w.h.p. $X = \Theta(n^2)$. ■

Let $i, j, k \in V$. We say that k is a *common neighbor* of i and j if $(k, i) \in E$ and $(k, j) \in E$.

Claim 5.8 *Let $i, j \in V$, and let Z be the number of their common neighbors. With high probability $Z < 3ln(n)$.*

Proof. W.l.o.g. let $i = 1, j = 2$. For $3 \leq k \leq n$ let $Z_k = 1$ if $(i, k) \in E$ and $(j, k) \in E$ and $Z_k = 0$ otherwise. Hence, $Z = \sum_{i=1}^{n} Z_i$ (we will add 2 more independent dummy variables Z_1, Z_2 to simplify the calculations) and $Pr(Z_k = 1) = \frac{1}{n}$. Therefore since $np = 1$, by Theorem 5.4 (1)

$$Pr[Z - 1 > 3ln(n)] < e^{3ln(n) - ln(1+3ln(n)) - 3ln(n)ln(1+3ln(n))} < e^{-3ln(n)} = \frac{1}{n^3}.$$

The last inequality is true since: $3ln(n) - ln(1 + 3ln(n)) - 3ln(n)ln(1 + 3ln(n)) < 3ln(n) - 3ln(n)ln(1 + 3ln(n)) < -3ln(n)$ if $ln(1 + 3ln(n)) > 2$, *i.e.*, if $n > 1$. ■

Claim 5.9 *With high probability no pair of vertices $i, j \in V$ has more than $3ln(n)$ common neighbors.*

Proof. From the previous claim the probability for a pair i, j to have more than $3ln(n)$ common neighbors is less than $\frac{1}{n^3}$. Since there are $\binom{n}{2}$ pairs the probability that at least one of them has more than $3ln(n)$ such neighbors is less than $\binom{n}{2}\frac{1}{n^3}$, *i.e.,* the probability for this event is less than $\frac{1}{n}$. ■

Corollary 5.10 *With high probability no pair of vertices (i, j) is incident on more than $(3ln(n))^2$ common 4-cycles such that i, j are not neighbors in the 4-cycle .*

Theorem 5.11 *With high probability the random graph $G(n, \frac{1}{\sqrt{n}})$ has $O(n^{1.5})$ edges and its smallest chordal supergraph has $O(\frac{n^2}{log^2(n)})$ edges.*

Proof. Let us look at the graph $\hat{G}$ obtained by minimum chordal completion of a random graph $G(n, \frac{1}{\sqrt{n}})$. Every 4-cycle in $\hat{G}$ must have a chord. By Claim 5.7 w.h.p. $G(n, \frac{1}{\sqrt{n}})$ has $\Theta(n^2)$ 4-cycles (not necessarily induced), and by Corollary 5.10 w.h.p. an edge can be a chord of at most $9ln^2(n)$ 4-cycles. Therefore $\hat{G}$ has $\Omega(\frac{n^2}{9ln^2(n)})$ edges, so we must have added $\Omega(\frac{n^2}{log^2(n)})$ edges. ■

5.2.2 The minimum perfect graph containing a random graph

In this chapter we study the random graph $G(n, \frac{1}{n^{0.6}})$. We shall show that with high probability G has $O(n^{1.4})$ edges but the smallest perfect graph that contains it has $\Omega(n^{1.8})$ edges.

Claim 5.12 *For a random graph $G(n, \frac{1}{n^{0.6}})$, with high probability $m = O(n^{1.4})$.*

The proof is analogous to that of Claim 5.6.

Claim 5.13 *With high probability the number of labeled 5-cycles in G is $\Theta(n^2)$.*

Proof. The proof is along the lines of Claim 5.7: The number of possible labeled 5-cycles in a graph is $\frac{4!}{2}\binom{n}{5}$ (we have to choose the 5 vertices of the cycle and order them). For every 5-cycle we will define an indicator X_i, $0 < i \le \frac{4!}{2}\binom{n}{5}$ with $X_i = 1$ if that 5-cycle is present in G and $X_i = 0$ otherwise. Define $X = \sum X_i$. By linearity of expectation we have $E[X] = \Theta(n^2)$. As shown in Claim 5.6, $Var[X] \le E[X] + \sum_{i\sim j} Pr[X_i \wedge X_j]$, where $X_i \sim X_j$ if X_i and X_j refer to different cycles and have common edges. X_i and X_j may have one, two or three common edges. There are $O(n^8)$ sets with one common edge and the probability for each such set is p^9, there are $O(n^7)$ sets with two common edges and the probability for each such set is p^8 and there are $O(n^6)$ sets with three common edges and the probability for each such set is p^7 .
Therefore

$$Var[X] = O(n^2) + O(n^8p^9) + O(n^7p^8) + O(n^6p^7) = O(n^{2.6}).$$

Again, using Equation (5.1), since $E[X] = \Theta(n^2)$ then with probability larger than $1 - \frac{1}{n}$, $|X - E[X]| \le \sqrt{nVar[X]} = O(n^{1.8})$, *i.e.*, $X = \Theta(n^2)$. ∎

Claim 5.14 *With high probability the degree of every vertex in G is at most $2n^{0.4}$.*

Proof. For $2 \le i \le n$ let $X_i = 1$ if $(1,i) \in E$ and $X_i = 0$ otherwise. Define $X = \sum_{i=1}^{n} X_i$ by Theorem 5.4 (3) we have

$$Pr(X - n^{0.4} > n^{0.4}) < e^{\frac{-2n^{0.4}}{27}} < \frac{1}{n^2}$$

(the last inequality is true for large enough n). Therefore the probability that there is a vertex with degree higher than $2n^{0.4}$ is less than $\frac{1}{n}$. ∎

Claim 5.15 *With high probability there is no pair of vertices $i, j \in V$ for which there exist $\Omega(n^{0.2})$ pairs of vertices $k, l \ne i, j$ such that $(i,k), (k,l), (l,j) \in E$.*

Proof. For a pair of vertices i, j let us look at the set $B_{i,j} = Adj(i) \cup Adj(j) \setminus \{i, j\}$. Since with high probability the degree of every vertex is less than $2n^{0.4}$, with high probability $|B_{i,j}| < 4n^{0.4}$ for all i, j. We now argue that the probability that $|E(G_{B_{i,j}})| = \Omega(n^{0.2})$ is less than $\frac{1}{n^3}$. To see this let us look at the graph $G(t = 4n^{0.4}, n^{-0.6})$. With probability larger than $1 - \frac{1}{n^3}$, G has $\Theta(\binom{t}{2}n^{-0.6}) = \Theta(n^{0.2})$ edges (the proof is similar to that of Lemma 5.6). The number of pairs k, l such that $(i, k), (k, l), (l, j) \in E$ is less than the number of edges in $G_{B_{i,j}}$, therefore with probability less than $\frac{1}{n}$ there is a pair i, j with $\Omega(n^{0.2})$ such paths. ■

Claim 5.16 *Let $X_{i,j}$ be the number common neighbors vertices of vertices i and j in V. With high probability $X_{i,j} < 20$ for all i, j.*

Proof. Let $X^k_{i,j} = 1$ if $(i, k) \in E$ and $(j, k) \in E$, and define $X_{i,j} = \sum X^l_{i,j}$,

$$Pr[X_{i,j} > C] = \sum_{i=C}^{n} \binom{n}{i} n^{-1.2i}(1 - n^{-1.2})^{(n-i)} \leq \sum_{i=C}^{n} n^i n^{-1.2i} =$$

$$= \sum_{i=C}^{n} n^{-0.2i} < n * n^{-0.2C} < \frac{1}{n^3}.$$

Therefore, with high probability for all i, j is $X_{i,j} < C$. ■

Claim 5.17 *With high probability there is no pair of vertices $i, j \in V$ that are contained in $\Omega(n^{0.2})$ 5-cycles such that i, j are not neighbors in those 5-cycles.*

Proof. Suppose i, j are contained in a 5-cycle such that i, j are not neighbors in this 5-cycle. Then there exist three vertices k, l, m such that $(i, k), (j, k) \in E$ and $(i, l), (l, m), (j, m) \in E$. By the previous claims there are at most $C \leq 20$ such k's and $O(n^{0.2})$ such pairs of l, m. Therefore there are $O(n^{0.2})$ such 5-cycles. ■

Theorem 5.18 *For the random graph $G(n, n^{-0.6})$, with high probability the smallest perfect graph containing G has $\Omega(n^{1.8})$ edges.*

Proof. Consider a minimal perfect graph G' containing G. Every 5-cycle in G' has a chord. From Claim 5.13 G has $\Theta(n^2)$ 5-cycles (not necessarily induced), and from Claim 5.17 an edge can be a chord of $O(n^{0.2})$ 5-cycles. Therefore G' has $\Omega(\frac{n^2}{n^{0.2}}) = \Omega(n^{1.8})$ edges, so we added $\Omega(n^{1.8})$ edges. ■

5.2.3 The minimum perfect deletion in a random graph

In this chapter we study the random graph $G(n, \frac{1}{2})$. We shall show that with high probability G has $\Theta(n^2)$ edges and the minimum perfect deletion set also contains $\Theta(n^2)$ edges. Let m be the number of edges in G.

Claim 5.19 *For the random graph $G(n, \frac{1}{2})$, with high probability $m = \Theta(n^2)$.*

Proof. Define a random variable e_i for each edge, $e_i = 1$ if $i \in E$ and $e_i = 0$ otherwise. Then $m = \sum_{i=1}^{\binom{n}{2}} e_i$. Since the variables are independent, and $p(e_i = 1) = \frac{1}{2}$, by Theorem 5.4 (2) $Pr[|m - \frac{\binom{n}{2}}{2}| > n\sqrt{n}] < e^{-\frac{n^3}{\binom{n}{2}}} < e^{-n}$. Therefore with high probability $\frac{\binom{n}{2}}{2} - n^{1.5} < m < \frac{\binom{n}{2}}{2} + n^{1.5}$. ■

Claim 5.20 *Let X be the number of induced 5-cycles in G. With high probability $X = \Theta(n^5)$.*

Proof. We assign an indicator X_i for every set of 5 vertices. Let $X_i = 1$ if the graph induced by these five vertices is an induced 5-cycle, and $X_i = 0$ otherwise. It can be easily checked that $Pr[X_i = 1] = \frac{4!}{2*2^{10}} = \frac{3}{256}$.

$X = \sum_{i=1}^{\binom{n}{5}} X_i$, and by linearity of expectation we have $E[X] = \frac{3}{256}\binom{n}{5} = \Theta(n^5)$.

$$Var[X] \leq E[X] + \sum_{i \sim j} Pr[X_i \wedge X_j]$$

In our case $i \sim j$ if X_i and X_j are different and have at least 2 common vertices. There are $O(n^8)$ sets with two common vertices, $O(n^7)$ sets with three common vertices and $O(n^6)$ set with four common vertices. The probability $Pr[X_i \wedge X_j]$ is less than 1, therefore $Var[X] = O(n^8)$. Using Equation (5.1), since $E[X] = \Theta(n^5)$, with probability larger than $1 - \frac{1}{n}$, $|X - E[X]| < \sqrt{nVar(X)} = O(n^{4.5})$ *i.e.,* $X = \Theta(n^5)$.

■

Claim 5.21 *Any edge in E belongs to $O(n^3)$ induced 5-cycles.*

Proof. Let $e_i \in E$. Each 5-cycle containing e_i as an edge contains only 3 more vertices and there is a constant number of possible 5-cycles on each set of 5 vertices. ■

Theorem 5.22 *With high probability the random graph $G(n, \frac{1}{2})$ has $\Theta(n^2)$ edges and minimum perfect deletion set of size $\Theta(n^2)$.*

Proof. Let us look at a minimum perfect deletion set of the random graph $G(n, \frac{1}{2})$. We have to remove an edge from every induced 5-cycle. Since there are $\Theta(n^5)$ induced 5-cycles and each edge can participate in at most $O(n^3)$ induced 5-cycles, we have to remove $\Omega(\frac{n^5}{n^3}) = \Omega(n^2)$ edges. ■

Bibliography

[1] A. Agrawal, P. Klein, and R. Ravi. Ordering problems approximated: single processor scheduling and interval graph completion. In *Proc. ICALP 1991*, pages 751–762. Springer, 1991. LNCS 510.

[2] A. Agrawal, 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.

[3] N. Alon, J. Spencer, and P. Erdos. *The Probabilistic Method*. Wiley, New York, 1991.

[4] A. Ben-Dor. Private communication, 1996.

[5] Amir Ben-Dor and Zohar Yakhini. Clustering gene expression patterns. In *Proc. RECOMB 99*, 1999.

[6] H. L. Bodlaender, M. R. Fellows, and T. J. Warnow. Two strikes against perfect phylogeny. In W. Kuich, editor, *Proc. 19th ICALP*, pages 273–283, Berlin, 1992. Springer. Lecture Notes in Computer Science, Vol. 623.

[7] H.L. Bodlaender and B. de Fluiter. On intervalizing k-colored graphs for DNA physical mapping. *Discrete Applied Math.*, 71:55–77, 1996.

[8] B. Bollobás. *Random Graphs.* Academic Press, 1985. Chapter XII.

[9] Andreas Brandstadt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes : A Survey.* Siam Monographs on Discrete Mathematics and Applications, 1999.

[10] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58:171–176, 1996.

[11] H. Chernoff. A measure of the asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[12] K. Cirino, S. Muthukrishnan, N.S. Narayanaswamy, and H. Ramesh. Graph editing to bipartite interval graphs: exact and asymptotic bounds. Technical report, Bell Laboratories Innovations, Lucent Technologies, 1996.

[13] M. R. Fellows, M. T. Hallet, and H. T. Wareham. DNA physical mapping: Three ways difficult. In *Proc. First European Symp. on Algorithms (ESA '93)*, pages 157–168. Springer, 1993. LNCS 726.

[14] 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.

[15] M. R. Garey, D. S. Johnson, and L. Stockmeyer. *Theor. Comput. Sci.*, 1:237–267, 1976.

[16] 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.

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

[18] M. C. Golumbic, H. Kaplan, and R. Shamir. On the complexity of DNA physical mapping. *Advances in Applied Mathematics*, 15:251–261, 1994.

[19] M. C. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. *Journal of Algorithms*, 19:449–473, 1995.

[20] M. C. Golumbic and R. Shamir. Complexity and algorithms for reasoning about time: A graph-theoretic approach. *J. ACM*, 40:1108–1133, 1993.

[21] S. Louis Hakimi, Edward F. Schmeichel, and Neal E. Young. Orienting graphs to optimize reachability. *Information Processing Letters*, 63(5):229–235, September 1997.

[22] 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.

[23] P. L. Hammer and B. Simeone. The splittance of a graph. *Combinatorica*, 1:275–284, 1981.

[24] J. Håstad. Some optimal inapproximability results. In *Proc. 29th STOC*, pages 1–10, 1997. Full version available as E-CCC Report number TR97-037.

[25] H. Kaplan and R. Shamir. Bounded degree interval sandwich problems. *Algorithmica*, 24:96–104, 1999.

[26] 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*, 28:1906–1922, 1999.

[27] H. Kaplan and M. Szegedi. Private communication, 1998.

[28] Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103, New York, 1972. Plenum Press.

[29] T. Kashiwabara and T. Fujisawa. An NP-complete problem on interval graphs. In *IEEE International Symposium on Circuits and Systems (12th)*, pages 82–83. Institute of Electrical and Electronics Engineers; Piscataway, N.J., 1979.

[30] J.M Lewis and M. Yannakakis. The node deletion problem for hereditary properties is NP-complete. *J. of Computer and System Sciences*, 20:219–230, 1980.

[31] L. Lovász. A characterization of perfect graphs. *J. Combin. Theory*, pages 95–98, 1972.

[32] 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, July 1993. Springer. LNCS 700.

[33] N.V.R. Mahadev and U.N. Peled. *Threshold Graphs and Related Topics*. Elsevier, North-Holland, 1995. Annals of Discrete Mathematics, Vol. 56.

[34] F. R. McMorris, T. J. Warnow, and T. Wimer. Triangulating vertex colored graphs. *SIAM J. Discrete Math.*, 7:296–306, 1994.

[35] 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.

[36] A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. In *Proc. 25th Int. Workshop (WG '99), Graph-Theoretic Concepts in Computer Science*. Springer-Verlag, 1999.

[37] 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.

[38] R. Sharan. Private communication, 1997.

[39] P. L. Tche'byshef. Des valeurs moyennes. *Journal de Mathematiques pures et appliquees*, pages 177–184, 1867.

[40] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming. In *Proc. IEEE Symposium on Foundations of Computer Science (FOCS '96)*, pages 617–626, 1996.

[41] M. Yannakakis. Node and edge deletion NP-complete problems. In *Proc. 10th STOC*, pages 253–264, 1978.

[42] M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM J. Alg. Disc. Meth.*, 2, 1981.