MATRIX PARTITION OF CHORDAL GRAPHS

by

Shekoofeh Nekooei Rizi

B.Sc., Computer Science Sharif University of Technology, 2007

Unfinished M.Sc., Computer Science

Amirkabir University of Technology (Tehran Polytechnic), 2008

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTERS OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY

Fall 2010

APPROVAL

Name: Shekoofeh Nekooei Rizi

Degree: Masters of Science

Title of Thesis: Matrix Partition of Chordal Graphs

Examining Committee: Dr. Joseph G. Peters
Professor, Computing Science
Simon Fraser University
Chair

Dr. Pavol Hell
Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Arthur L. Liestman
Professor, Computing Science
Simon Fraser University
Co-Supervisor

Dr. Binay Bhattacharya
Professor, Computing Science
Simon Fraser University
Examiner

Date Approved: Dec 17th, 2010

Abstract

Matrix partitions were introduced by Feder, Hell, Klein and Motwani [18] as a common generalization of graph colourings, homomorphisms, and many partitions that play a role in the study of perfect graphs etc., [51, 50, 44, 10, 43, 6].

Many results were obtained in the literature, focusing on the complexity of the problem, depending on the matrices with special structures and restricted classes of graphs [15, 17, 20]. These results prove some problems are polynomial and some NP-complete. For the special case of the homomorphism problems, it is expected that all problems are polynomial or NP-complete [22]. In general it is not clear whether such a dichotomy is to be expected.

Another direction in the literature is classifying the matrix partition problems according to whether they have a finite or an infinite number of minimal forbidden subgraphs (minimal obstructions). There are results for general graphs, and for special classes of matrices [19, 16].

In this thesis we survey all these results, and we also make a contribution to the second classification (finite versus infinite number of minimal obstructions) for chordal graphs. While these classification is not complete, even for 3 by 3 matrices, we classify certain subset of 3 by 3 matrices (matrices with similar entries on the main diagonal). These results indicate the kind of methods that are useful in the classification and possibly suggest how a general classification may look.

Keywords: generalized graph colouring, matrix partition, chordal graphs, minimal obstructions, first order definable

دی شیخ با چراغ همی گشت گرد شهر
کز دیو و دد ملولم و انسانم آرزوست

گفتند یافت می نشود گشته ایم ما
گفت آنچه یافت می نشود آنم آرزوست

مولانا

Acknowledgments

It is my pleasure to express my profound gratitude to my supervisor, Dr. Pavol Hell. Not only does he hold expert command of the subject of Graph Theory, but he communicates this knowledge with great skill, enthusiasm, without which this work would never have been possible.

I would like to thank Bradley Coleman, my grad buddy at SFU, and Craig Weidert for their absolute help when I just arrived to the new country, and did not know anyone at all. I would very much appreciate Hossein Maserrat and Lisa Neame favours who generously welcomed me to live at their places for the first month that I got here, and did not have anywhere to stay.

I am also grateful for the advice and help of Hengameh and Phuong for designing the figures in the thesis. I would thank all of my supportive friends, classmates, labmates and roomates (back home and here), to whom I am very grateful, particularly Nasim, Razieh, Zeinab, Amir, Cida, Ehsan, Faraz, Fatemeh, Hengameh, Hoda, Kathryn, Laurens, Maryam, Mozhdeh, Murray, Naghmeh, Sahar, Samira, Shabnam, Sebastian, Shahab, Sia.

This course of study and the big step of my life have never been made and more enjoyable without the encouragement of my family members and relatives: my brother, grand mothers, aunts, uncles and cousins.

Finally, I would like to offer my endless gratitude to my mother and father for their unfailing kindness, and also their financial support whenever I needed. No matter how challenging the life would be for themselves, they would always love their children to have the bests.

Contents

List of Figures

Chapter 1

Introduction

In this chapter we define the terminology, notation, and some of the problems and concepts that we need for our research approach.

1.1 Basic Definitions

A *digraph* G consists a set of vertices, $V(G)$, and a set of edges, $E(G) \subseteq V(G) \times V(G)$. If $e = uv \in E(G)$ is an edge for $u, v \in V(G)$, we say that u dominates v. This edge e is directed from u to v. While we discuss digraphs occasionally, the focus of the thesis will be on graphs and the next few definitions treat the case of graphs. A *graph* is a special case of a digraph where $uv \in E(G)$ if and only if $vu \in E(G)$. (Thus in a graph, edges have no directions.) In this case we also say u and v are *adjacent*.

A *matching* in a graph is a set of edges without common vertices. An edge with the same endpoints is called a *loop*. A graph is *reflexive* if each vertex has a loop. A graph is *simple* when it has neither a loop nor multiple edges between the same vertices. The *degree* of a vertex v is denoted by $d(v)$ and indicate the number of edges adjacent to v, while loops considered twice. The maximum degree of a graph G, denoted by $\Delta(G)$.

A *subgraph* of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Note that if $u, v \in V(H)$ are adjacent in H they are also adjacent in G. An *induced subgraph* of graph G, is a subgraph H of G such that $u, v \in V(H)$ are adjacent in H if and only if they are adjacent in G. A *proper* induced subgraph H of a graph G is an induced subgraph H such that the set of vertices of H is a proper subset of vertices of G, i.e., $V(H) \neq V(G)$. The complement of a graph G is a graph $\bar{G}$ with $V(G)$ as the set of vertices; two distinct vertices u, v are adjacent in $\bar{G}$ if and only if u, v are not adjacent in G.

The following are the definitions of some of the graphs that we use frequently.

A *path* is an ordered sequence of distinct vertices $v_1, v_2, \cdots, v_n$ such that for $i = 1, \cdots, n-1$ the vertices v_i and v_{i+1} are adjacent and other pairs are non-adjacent. We use P_n to denote as a path with n vertices. A *cycle* is a path such that the vertices in the sequence are distinct, except that the first vertex is the same as the last vertex. A cycle of length n has n edges (vertices); it is called n-cycle and we use the notation C_n for it. An *odd cycle* is a cycle C_n such that n is an odd number. A *triangle* is a cycle of length three. A *pentagon* is a cycle of length five. A *forest* is a graph without an induced cycle. A *tree* is a forest such that there exists a path between every pair of its vertices. A *bipartite* graph is a graph without a cycle of odd length.

A *clique* is a graph such that all pairs of vertices are adjacent to each other. A clique is also called a *complete graph*, since all possible edges are present. A clique of *size* n has n vertices and is represented by K_n. A *maximal clique* of a graph is an induced subgraph which is a clique and cannot be expanded by adding one more vertex, such that the new expanded graph is also a clique.

A graph is called *connected* if there is a path between any two vertices. A *component* of a graph G is a maximal non-empty connected subgraph of G. (We sometimes say *connected component.*)

Next we explain some of the useful functions in graph theory; followed by some of the well-known graph problems.

An *independent set* of a graph is a subset of vertices which does not induce any edge. We say that two sets of vertices A, B are independent if there is no edge with one endpoint in A and the other one in B, i.e, there are no edges between A and B. Two edges are independent if they have different endpoints. (Two vertices are independent if they are not adjacent.) We let $\alpha(G)$ denote the size of the largest independent set in G. We let $\omega(G)$ denote the size of the largest clique in G. We use $\theta(G)$ to denote clique number of G, which is the fewest number of cliques that contain all the vertices of G. An *edge cover* of a graph G is a set of edges C such that each vertex is adjacent to at least one edge in C. For the input graph G, we use $\beta'(G)$ to denote the *edge cover number* of G which is the smallest possible size of and edge covers of G.

A *colouring* of a graph G is a function f that maps vertices of G to a set of colours C, such that for two adjacent vertices u and v, we have $f(u) \neq f(v)$. (A colouring maps adjacent vertices to different colours.)

The first problem we discuss is the well-known k-colouring problem.

A *k-colouring* is a colouring of G with a set of colours C of size k. (So a k-colouring uses at most k colours.) Given a graph G and a fixed positive integer number k, the *k-colouring problem* asks if G has a k-colouring. When $k = 1$, the problem is trivial: input graph G is 1-colourable if and only if it has no edges. When $k = 2$, the problem is also easy: graph G is 2-colourable if and only if it is bipartite, i.e., if and only if it has no odd cycles.

Theorem 1.1.1. *[39] The graph G has 2-colouring if and only if it is bipartite.*

□

Whether an input graph is bipartite, can be decided by an obvious Breadth First Search, therefore the problem of 2-colouring is polynomial in the size (the number of vertices plus the number of edges) of the input graph.

In the rest of this thesis when we say a problem for the input graph G is polynomial, we mean polynomial in the size of the number of vertices plus the number of edges of G.

Theorem 1.1.2. *[37] The k-colouring problem of an input graph G is polynomial if $k \leq 2$; it is NP-complete if $k \geq 3$.*

This theorem shows that we have a dichotomy for the k-colouring problem.

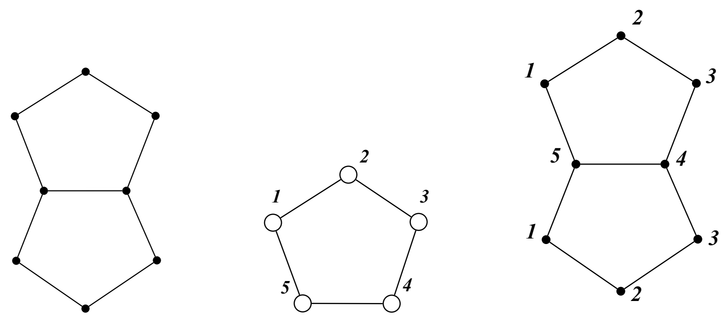
The *chromatic number* of G, denoted $\chi(G)$, is the smallest integer k such that there exists a k-colouring for G. Note that the chromatic number is also the smallest number of independent sets in any partition of $V(G)$.

The second problem we discuss is the graph homomorphism problem, that generalizes the graph colouring problem.

Let G and H be graphs. A *homomorphism* of G to H is a mapping $f : V(G) \to V(H)$ such that $xy \in E(G)$ implies $f(x)f(y) \in E(H)$. If there is a homomorphism from G to H, we say G is *homomorphic* to H, and we write $G \to H$. When G is not homomorphic to H we write $G \not\to H$. The *homomorphism problem* (for a fixed graph H) asks if the input graph G is homomorphic to H.

Here are two examples to illustrate the concept of the homomorphism problem.

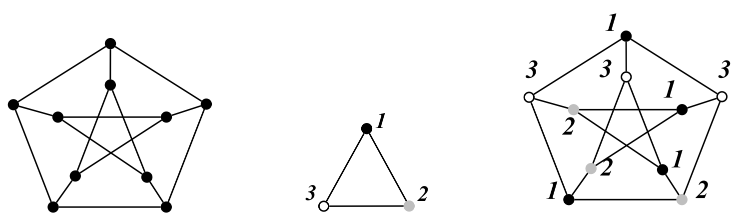
First Example: Suppose G is the graph shown in Figure 1.1(a) and H is a pentagon. A homomorphism of G to $H = C_5$ is shown in Figure 1.1(c)



(a) G (b) C_5 (c) $G \to C_5$

Figure 1.1: First example of homomorphism

Second Example: Suppose G is the *Petersen graph* which is shown in Figure 1.2(a) A homomorphism of G to $H = K_3$ is shown in Figure 1.2(c) .



(a) G =Petersen Graph (b) K_3 (c) $G \to K_3$

Figure 1.2: Second example of homomorphism

The homomorphism problem is a generalization of colouring problem, since if we let H be a complete graph, then we obtain the colouring problem. For instance the 3-colouring problem is indeed the homomorphism problem to K_3. Therefore the k-colouring problem is the K_k-homomorphism problem.

The following theorem shows a connection between two homomorphic graphs and their chromatic number.

Theorem 1.1.3. *If* $G \to H$ *then* $\chi(G) \leq \chi(H)$

Proof. Suppose G is homomorphic to H, and f is the homomorphism of G to H, and g is a colouring of H using $k = \chi(H)$ colours. Then the function g' defined as $g'(u) = g(f(u))$

is a colouring of G. Suppose two vertices u and v are adjacent in G, then $f(u), f(v)$ are also adjacent in H, since f is homomorphism of G to H. So $g(f(u)) \neq g(f(v))$, as g is a colouring of H. Therefore $g'(u) \neq g'(v)$. Obviously g' does not use more colours than g does, therefore $\chi(G) \leq k$. $\square$

The homomorphism problem is trivial when H has a loop. In this case, a homomorphism is obtained by mapping all the vertices to the looped vertex. This mapping is a homomorphism because any two adjacent vertices are mapped to a vertex that is adjacent to itself. Therefore every graph is homomorphic to a graph that has a loop.

The homomorphism problem is polynomial when H is a bipartite graph. Indeed, if the input graph G is bipartite, and $u, v \in V(H)$ are adjacent, then a homomorphism of G to H is obtained by mapping the vertices of one part to u and vertices of the other part to v. On the other hand, if the input graph G is not bipartite then $\chi(G) \geq 2$; but as H is bipartite, then $\chi(H) \leq 2$. Therefore, $\chi(H) \leq \chi(G)$ and Theorem 1.1.3 implies that G is not homomorphic to H.

It has been shown [36], cf. Chapter 5 of [35], that in all other cases, i.e., when H is not bipartite and has no loop, the homomorphism problem is NP-complete. Therefore we also have a dichotomy for the homomorphism problem of undirected graphs.

Theorem 1.1.4. *[36] If H is a bipartite graph or a graph with a loop, then the homomorphism problem to H is polynomial solvable. If H is non-bipartite and does not have loop, then the homomorphism problem to H is NP-complete.*

1.2 Forbidden Subgraphs

An *isomorphism* of a graph G to H is a homomorphism f of G to H that is bijective. (A bijective function is one-to-one and onto.) We say G is *isomorphic* to H if there is an isomorphism from G to H.

Let H be a fixed graph. A graph G is called *H-free* if it does not have an induced subgraph isomorphic to H. If $\mathcal{H}$ is a set of graphs, we say G is $\mathcal{H}$-free if G does not contain any of the members of $\mathcal{H}$. When $\mathcal{G}$ a class of graphs are $\mathcal{H}$-free, we say that each member of $\mathcal{H}$ is a *forbidden subgraph* for $\mathcal{G}$. We say that a graph H is a *minimal* forbidden subgraph for G, if H is a forbidden subgraph for G, but any proper induced subgraph of H, is not a forbidden subgraph of G.

For example let $\mathcal{T}$ be the family of trees. Then K_3 and K_4 are forbidden subgraphs for $\mathcal{T}$, however K_4 is not a minimal forbidden subgraph, but K_3 is minimal.

Some graph classes can be defined according to a set of minimal forbidden subgraphs. For example trees are connected graphs that do not induce a cycle; bipartite graphs are graphs that do not induce an odd cycle; and chordal graphs, that we study in Section 2.2, are graphs that do not induce cycles of length greater than three. All the graph classes in these examples have infinitely many minimal forbidden subgraphs. But cographs that we study in Section 2.2.4 have finitely many minimal forbidden subgraphs. Indeed cographs are a class of graphs that do not induce P_4 (path of four vertices).

Chapter 2

Graph Classes

In this chapter we will survey some of the well-known graph classes. Most of these classes will be used when we discuss the matrix partition problem for restricted graph classes.

2.1 Perfect Graphs

Most of the classes we discuss are subclasses of the class of perfect graphs, hence we discuss this class first.

For any graph G we always have $\omega(G) \leq \chi(G)$. This inequality holds since each vertex of a maximum clique in G must receive a different colour in a colouring of G. Therefore the chromatic number of G is at least the size of the maximum clique. Those graphs G for which these two values are equal, both for G and for all its induced subgraphs, make one of the most interesting and well behaved classes of graphs. One reason for their appeal is the equality of a certain minimum function (namely $\chi(G)$) with a certain maximum function (namely $\omega(H)$). Thus the perfectness of a graph (or family of graphs) represents a theorem equating a certain minimum to a certain maximum, i.e., a min-max result.

A graph G is called *perfect* if for every set of vertices $V' \subseteq V$, the induced subgraph G' of G over V' satisfies $\omega(G') = \chi(G')$.

The importance of perfect graphs was recognized by Claude Berge who noticed that many well-known min-max results such as König's Theorem, Dilworth's Theorem, etc., correspond to the perfectness of natural classes of graphs [1, 2]. We will discuss these classes in the rest of this section; they include bipartite graphs, their complements, line-graphs of bipartite graphs, and their complements. Also, in the next section we will discuss chordal graphs, comparability graphs, interval graphs, cographs, and split graphs which are subclasses of perfect graphs.

It is known that bipartite graphs are perfect. In the following we discuss an argument for this result. We first show that bipartite graphs are perfect. Obviously, if a bipartite graph G does not have any edges, the equality holds since $\omega(G) = 1$ and $\chi(G) = 1$; and as any subgraph of G is also a bipartite graph without edges, the equation holds for every induced subgraph of G. So, consider a bipartite graph G, that has at least an edge. The chromatic number of G is two (bipartite graphs have a 2-colouring, two colours are needed since G has at least two adjacent vertices), thus $\chi(G) = 2$. On the other hand as G does not induce a cycle (specifically a $C_3 = K_3$), its clique number is two, i.e., $\omega(G) = 2$. So G has the equality: $\chi(G) = \omega(G)$. We know that each induced subgraph G' of G is also bipartite, so whether or not G' has an edge the equation $\omega(G') = \chi(G')$ holds. As the result, bipartite graphs are perfect.

We also know that complements of bipartite graphs are perfect. Recall that $\alpha(G)$ denotes the independence number of G; and $\beta'(G)$ denotes the edge cover number which is the minimum number of edges that contain all the vertices of G.

Theorem 2.1.1. *(König 1916 [39]) If a graph G is bipartite without isolated vertices, then $\alpha(G) = \beta'(G)$.*

This theorem deals with bipartite graphs without isolated vertices. But a graph G with isolated vertices, the function $\beta'(G)$ is not defined. However when we talk about perfectness of graphs we must consider all induced subgraphs, and there might be isolated vertices in some induced subgroups. Thus we shall also consider bipartite graphs with isolated vertices. Recall that $\theta(G)$ is the clique cover of G, which is the fewest number of cliques that cover (contain) all vertices of G. Note that in bipartite graphs each clique is a K_2 (an edge) or a K_1 (an isolated vertex). The following corollary states that the independence number and the clique cover number for bipartite graphs are equal.

Corollary 2.1.1. *[27] If a graph G is bipartite, then $\alpha(G) = \theta(G)$.*

Proof. Let V' be the set of isolated vertices of a bipartite graph G, and $G' = G \backslash V'$ (which is a bipartite graph without isolated vertices.) It is easy to see that $\theta(G) = \beta'(G') + |V'|$ Indeed, each clique in the clique cover of G' is a K_2, and is equivalent to an edge in the edge cover of G'. The clique cover of G includes these edges as well as the isolated vertices. Also it is trivial to see that $\alpha(G) = \alpha(G') + |V'|$. Theorem 2.1.1 implies that $\alpha(G') = \beta'(G')$. As we explained $\theta(G) = \beta'(G') + |V'|$ and $\alpha(G) = \alpha(G') + |V'|$; therefore in any bipartite graph G the equation $\alpha(G) = \theta(G)$ holds. $\square$

For any graph G we have: $\alpha(G) = \omega(\bar{G})$ and $\theta(G) = \chi(\bar{G})$. The latter equation holds because: vertices in each clique of the clique cover of G are independent in $\bar{G}$, so they can have the same colour in a proper colouring of $\bar{G}$; therefore $\chi(\bar{G}) \leq \theta(G)$. On the other hand vertices of different cliques in G are adjacent in $\bar{G}$, so they must receive different colours in a proper colouring of $\bar{G}$, so any colouring of $\bar{G}$) needs at least $\theta(G)$ colours, i.e., $\theta(G) \leq \chi(\bar{G})$. As a result we have $\theta(G) = \chi(\bar{G})$, for any graph G. If G is a bipartite graph, using Corollary 2.1.1 and putting these equations together, we have $\omega(\bar{G}) = \chi(\bar{G})$. The same arguments apply for all induced subgraphs, because each induced subgraph of $\bar{G}$ is the complement of some induced bipartite graph. As the result the complement of bipartite graphs are perfect.

Another natural subgraph of perfect graphs is the class of line-graphs of bipartite graphs. The line-graph of a graph G is a graph that represents the adjacencies between edges of G. Formally the *line-graph* of a graph G, written $L(G)$, is a graph with the set of vertices $V(L(G)) = E(G)$ and two distinct vertices e, f are adjacent in $L(G)$ if and only if the edges e, f have a common endpoint in G. For the line-graph of a bipartite graph G we have the equality $\Delta(G) = \omega(L(G))$. Indeed if a vertex v in G has d adjacent edges, then these edges in $L(G)$ make a clique of size d and vice versa (except when $d = 3$, a triangle in G yields a triangle in $L(G)$). Therefore the maximum clique in L has the same size as the maximum degree of the vertices in G. A proper colouring of $L(G)$ is a proper edge colouring of G, therefore $\chi(L(G)) = \chi'(G)$. Using the following theorem, we can put these equations together.

Theorem 2.1.2. *(König 1916 [39]) If G is a bipartite graph, then $\chi'(G) = \Delta(G)$.*

Thus for the bipartite graph G, the line-graph $L(G)$ satisfies the equation $\omega(L(G)) = \chi(L(G))$. The same arguments show that the equation holds for each induced subgraph of $L(G)$ as well. So line-graphs of bipartite graphs are perfect.

Finally, it is known that the complements of line-graphs of bipartite graphs are perfect. Here we argue why this result holds. Consider a graph G. A clique of size k in $\bar{L}(G)$ is a disjoint set of edges in G of size k. Therefore the clique size of $\bar{L}(G)$ is equal to the size of the maximum matching in G. On the other hand, suppose a proper colouring for $\bar{L}(G)$. Vertices with the same colour form a clique in $L(G)$, and a set of edges with one common endpoint in G. Therefore a proper colouring in $\bar{L}(G)$ is equivalent to a set of vertices V in G, such that each edge of G has an endpoint in V (so V is a vertex cover). So $\chi(\bar{L}(G))$ is equal to the minimum vertex cover of G.

Theorem 2.1.3. *(König-Egerváry 1931 [40, 14]) In any bipartite graph G, the number of edges in a maximum matching equals the number of vertices in a minimum vertex cover.*

So, in bipartite graph G the minimum vertex cover and the maximum matching have equal size, therefore $\omega(\bar{L}(G)) = \chi(\bar{L}(G))$. As the result, complements of line-graphs of bipartite graphs are perfect.

2.1.1 Characterizations of Perfect Graphs

In 1960, Berge [1] made two conjectures for perfect graphs. One of the conjectures was proved twelve years later, by Lovász; but it took over forty years to prove the other conjecture.

The first conjecture was motivated by the phenomenon we have already observed - namely that complements of perfect graphs also tend to be perfect.

Theorem 2.1.4. *(Lovász 1972 [43]) A graph is perfect if and only if its complement is perfect.*

Next we describe two important classes of minimal graphs that are not perfect.

Consider an odd cycle C_{2k+1} with the set of vertices $V = \{v_1, \cdots, v_{2k+1}\}$. The chromatic number of the odd cycle is three. Since the odd cycles are not bipartite the chromatic number is not two; a 3-colouring of C_{2k+1} can be obtained by colouring the even index vertices by the first colour and the odd index vertices, except the last one, by the second colour, and v_{2k+1} by the third colour. On the other hand the maximum clique of an odd cycle of length greater than three is two. Therefore $\chi(C_{2k+1}) > \omega(C_{2k+1})$ when $k > 1$. Hence odd cycles are not perfect.

This inequality also holds for the complement of odd cycles. For complement of an odd cycle C_{2k+1} we have $\chi(\bar{C}_{2k+1}) = k+1$; In a colouring of C_{2k+1} it is not possible to have three or more vertices with the same colour, because each vertex is non-adjacent to two vertices, but those two vertices are adjacent; therefore at least $\lceil \frac{2k+1}{2} \rceil = k+1$ colours are needed for a colouring of a $\bar{C}_{2k+1}$. A $(k+1)$-colouring of $\bar{C}_{2k+1}$ can be obtained by colouring v_{2i-1} and v_{2i} by the i^{th} colour for $i = 1, \cdots k$, and the last vertex by $k+1^{th}$ colour. The clique number of $\bar{C}_{2k+1}$ is k. The maximum clique can be obtained by an induced graph on v_{2i-1} for $i = 1, \cdots, k$. So we have $\chi(\bar{C}_{2k+1}) > \omega(\bar{C}_{2k+1})$ which means that the complements of odd cycles are not perfect either.

Note that induced subgraphs of perfect graphs are also perfect, by the definition. Hence any graph containing an odd cycle or a complement of an odd cycle is not perfect. The second conjecture made by Berge states that all other graphs are perfect.

Theorem 2.1.5. *(Chudnovsky et. al 2003 [6]) A graph G is perfect if and only if G and $\bar{G}$ do not have an induced odd cycle of length greater than three.*

We note that it follows from the proof of Theorem 2.1.5 in [6] that all perfect graphs can be constructed from bipartite graphs, their complements, line-graphs of bipartite graphs, and their complements, by some basic operations. Each of the operations corresponds to a particular kind of the matrix partition problem.

The proof in [6] has also been enhanced to produce a polynomial algorithm to decide whether a given graph G is perfect or not [5].

Finally we note that for perfect graphs G, all parameters $\alpha(G)$, $\omega(G)$, $\chi(G)$, and $\theta(G)$ can be computed in polynomial time [29].

2.2 Chordal Graphs and Subclasses

A graph G is *chordal* if every cycle of length strictly greater than three has an edge joining two non-adjacent vertices of the cycle; such an edge is called *chord*. In other words, a chordal graph is a graph that does not contain an induced n-cycle for $n > 3$. We see that any induced subgraph of chordal graph G also does not induce any cycle of length greater than three, thus also is chordal.

2.2.1 Characterizations of Chordal Graphs

Using Theorem 2.1.5, it is easy to see that chordal graphs are perfect. However, we will show this directly from the definition by proving that a chordal graph G has $\omega(G) = \chi(G)$.

(Since induced subgraphs of chordal graphs are chordal, this will suffice.)

First, we mention some characterization of this class of graphs.

A vertex v in graph G is called *simplicial* if its neighbours induce a complete subgraph in G.

Theorem 2.2.1. *(Dirac 1961 [12]) A chordal graph always has a simplicial vertex.*

A *perfect elimination ordering* (sometimes called *simplicial ordering*) in a graph G is an ordering of the vertices of the graph $v_1, \cdots, v_n$ such that, for each vertex v_i, the forward neighbours of v_i (the neighbours of v_i among $v_{i+1}, \cdots, v_n$) form a clique in G. So, in a chordal graph G, any three vertices v_i, v_j, v_k such that $i < j < k$ and $v_iv_j \in E(G)$ and $v_iv_k \in E(G)$ implies that $v_jv_k \in E(G)$.

Theorem 2.2.2. *(Fulkerson-Gross 1965 [23]) A graph is chordal if and only if it has a perfect elimination ordering.*

Proof. Suppose G is a chordal graph. Theorem 2.2.1 implies that G has a vertex v whose neighbours induce a clique. We put v as the first vertex of the ordering, $v = v_1$, and recursively consider the graph $G = G \backslash \{v\}$ which is still a chordal graph. In this way we obtain a perfect elimination ordering.
To prove the converse assume G has a perfect elimination ordering but is not chordal, i.e., has an induced cycle C of length at least four. Then let v be the vertex of C that appears first in the ordering, and let u and w be its neighbours in C. Since u, w appear after v they must be adjacent, which is not the case in C. $\square$

Theorem 2.2.3. *(Berge [2]) Chordal graphs are perfect.*

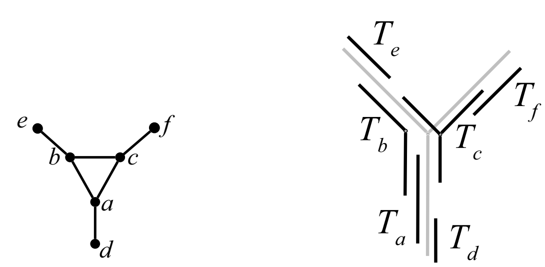
Proof. Assume G is a chordal graph, and $v_n, \cdots, v_1$ is the reverse of the perfect elimination ordering; so for each v_i the neighbours of v_i in $\{v_n, \cdots, v_{i+1}\}$ form a clique. Starting from v_n, we apply the smallest possible colour (suppose colours are integers). If vertex v_i receives the k^{th} colour, then each of the first $k - 1^{th}$ colours must appear in some neighbour of v_i among $v_n, \cdots, v_{i+1}$. Since these vertices together with v_i form a clique, we have a clique of size k. In this way the minimum number of colours we need to have a proper colouring of G is equal to the largest clique of G. $\square$

Corollary 2.2.1. *[27] The graph colouring and the clique problem are polynomial for chordal graphs.*

Theorem 2.2.4. *[23] Any chordal graph G with n vertices has at most n maximal cliques.*

Proof. Consider the perfect elimination ordering $v_1, \cdots, v_n$ for the vertices of a chordal graph G. For each $v_i \in V(G)$ in the ordering, define $F(v_i)$ as the induced subgraph on v_i and its forward neighbours in the ordering. The perfect elimination ordering property implies that $F(v_i)$ is a clique. We call $F(v_i)$ the forward clique for v_i. Consider a maximal clique in G and suppose v_t is the vertex with the smallest index in this clique which appears in the ordering. Other vertices in this maximal clique are neighbours of v_t and have greater indexes. So they are forward neighbours of v_t in the ordering. Therefore the maximal clique is $F(v_t)$. As for each maximal clique there is one forward clique, so for a graph with n vertices there are at most n maximal cliques. □

Another property of chordal graphs is their intersection representation. A graph G is chordal if we can put its vertices $v \in V(G)$ in a one-to-one correspondence to a set of subtrees $\{T_v\}_{v\in V(G)}$ on a tree, so that two vertices are adjacent if and only if the subtrees intersect. You can see an example in Figure 2.1.



(a) G (b) T_G

Figure 2.1: A subtree intersection of the chordal graph G is T_G .

Theorem 2.2.5. *(Gavril 1974 [24]) A graph is chordal if and only if it is the intersection graph of a family of subtrees of a tree.*

Recognition of Chordal Graphs

As noted above any subgraph of a chordal graph is also chordal, and any chordal graph has a simplicial vertex. Using a brute-force algorithm, we find a simplicial vertex v, and put it in a queue; then delete v and repeat the same thing for the rest of the graph.

A faster approach was presented by Rose, Tarjan, Lueker in 1976 [47]. It uses Lexicographical Breadth First Search (LBFS). The idea for their algorithm is that a perfect elimination ordering can end at any vertex. Therefore the algorithm generates the perfect elimination ordering backwards. It starts at any vertex of G, and the vertices are numbered from n to 1, indicating the position of each vertex in the ordering. The algorithm also maintains a $l$abel at each vertex. The label is a string. In the beginning all the vertices have a null label. The labels are updated by adjoining the number v at the end each iteration. So for each vertex u, the label at u is a sequence of numbers in decreasing order. The labels are comparable in lexicographic order.

The algorithm always produces an ordering of vertices, even if a graph is not chordal. So it must be checked if it is perfect elimination ordering to decide the input graph is chordal.

Theorem 2.2.6. *[27] The input graph G is chordal if and only if the ordering produced by LBFS is a perfect elimination ordering. It can recognize if a graph G is chordal in $O(|V(G)| + |E(G)|)$ time complexity.*

Another method by Tarjan (1976) can also be used for recognizing chordal graphs. It is an unpublished lecture note [49]. This method is called Maximum Cardinality Search (MCS). The vertices are numbered starting from n down to 1. The next vertex to be numbered is always a neighbour of the largest numbered vertices. Ties are arbitrarily broken.

Corollary 2.2.2. *[49] Chordal graphs can be recognized in linear time in the size of the input graph.*

2.2.2 Comparability Graphs

Another subclass of perfect graphs is the class of comparability graphs that we briefly study here. A *transitive orientation* of a graph G is an assignment of exactly one direction to each edge of G, such that if there is an edge oriented form u to v and an edge from v to w

then there is an edge from u to w. Note if G does not have any loops, then its transitive orientation cannot have any directed cycle.

A graph G is called a *transitively orientable* graph or *comparability* graph, if there exists a transitive orientation for it. An example of a transitive orientation is shown in Figure 2.2.

(a) (b)

Figure 2.2: Transitive orientation of a graph known as the butterfly graph.

Theorem 2.2.7. *(Berge 1960 [1]) Comparability graphs are perfect.*

Proof. It is easy to see that any induced subgraph of a comparability graph is also a comparability graph. Using this fact it is enough to show that for a comparability graph G, we have $\chi(G) = \omega(G)$. Let D be the graph which is obtained by transitive orientation of G. As D does not have a directed cycle, for each vertex v, we can define $\ell(v)$ as the length of the longest directed path that ends in v. A colouring of vertices of G that colours each vertex v by colour $\ell(v)$ is a proper colouring for G. Because if u and v are adjacent and the oriented edge is from u to v then $\ell(v) \geq \ell(u) + 1$. Therefore $\chi(G) \leq \omega(G)$ and as we already mentioned for each graph we have $\omega(G) \leq \chi(G)$. So for a comparability graph G we have $\chi(G) = \omega(G)$, hence it is perfect. $\square$

Each transitively oriented graph defines a natural partial order where $u \prec v$ if there is a directed path from u to v. Recall that an *antichain*, in a partially ordered set, is a set of elements no two of which are comparable to each other; and a *chain* is a set of elements every two of which are comparable. Then in a transitive graph, each antichain is an independent set and each chain is a clique, and we conclude that for comparability graphs G we also have $\alpha(G) = \theta(G)$. Dirac has shown that in a partial order the maximum size of an antichain equals the minimum number of chains covering all elements. Putting these facts together, we have the following fact.

Corollary 2.2.3. *(Dirac [12]) The complement of comparability graphs are perfect.*

A transitive orientation of a graph, if it exists, can be found in linear time [45]. However, the algorithm for doing so will assign orientations to the edges of any graph; then we need to test whether the resulting orientation is transitive, which is equivalent the complexity of matrix multiplication, and currently the best algorithm for it is of $O(n^{2.367})$ [53].

2.2.3 Interval Graphs

The class of interval graphs is one of the principal graph classes, and one the best-known subclasses of chordal graphs that has been introduced and studied. We survey this class briefly, even though it does not enter into the next chapters.

A graph G is called an *interval graph* if its vertices $v \in V(G)$ can be put in a one-to-one correspondence to a set of closed intervals $\{I_v\}_{v \in V(G)}$ in the straight line, such that u, v are adjacent if and only if their corresponding intervals I_u, I_v have a non-empty intersection. The set of intervals $\{I_v\}_{v \in V(G)}$ is called an *interval representation* of G.

Note that interval graphs are reflexive, i.e., each vertex has a loop since each interval intersects itself in the interval representation.

Some Characterization of Interval Graphs

It is easy to see that an induced subgraph of an interval graph is an interval graph. (Given an interval representation of the graph we obtain an interval representation of the subgraph, by keeping only the intervals corresponding to the vertices of the subgraph.)

A cycle of length four or more does not have an interval representation. Indeed, suppose we have an interval representation of a cycle c of length $k \geq 4$. Let I_v be the interval with the least left endpoint representing a vertex v of the cycle. Let u and w be the neighbours of v on the cycle c. We claim that both the intervals corresponding to u and w must contain the right endpoint of I_v. (Otherwise the vertices u, w could not have neighbours that are not adjacent to v.) But this means that u and w are adjacent in G. Therefore, an interval graph can not contain an induced cycle of length greater than three. We have shown the following fact.

Theorem 2.2.8. *[26] Each interval graph is chordal (and hence perfect).*

Not all chordal graphs are interval graphs. There is another forbidden structure for interval graphs.

An *asteroidal triple* of a graph consist of three nonadjacent vertices u, v, w and, for each pair, a path joining them without containing a neighbour of the third vertex. In other words a path P_{uv} joins u and v and does not contain a neighbour of w, a path P_{uw} joins u and v and does not contain a neighbour of v; and similarly a path P_{vw} joins v and w and does not contain a neighbour of u.

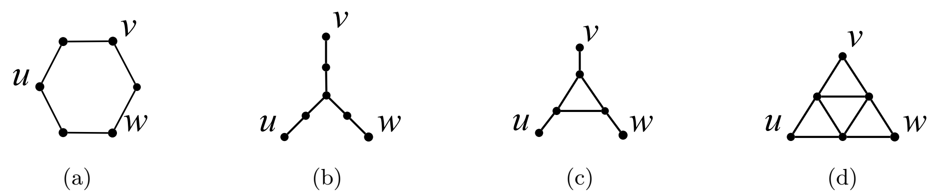


Figure 2.3: Example graphs with asteroidal triples.

Theorem 2.2.9. *[42] An interval graph does not have an asteroidal triple as an induced subgraph.*

Proof. Suppose we have an interval representation of an asteroidal triple graph; and u, v, w are three independent vertices; with I_u, I_v, I_w as their corresponding intervals, respectively. Not two of these intervals intersect. So without loss of generality we suppose that they are ordered as I_u, I_v, I_w on the straight line (from left to right). By definition, there must be a path from u to w that does not contain a neighbour of v. But if in the interval representation it means there must be a sequence of interval between I_u and I_w, pairwise intersecting but without intersecting the interval I_v which is in between. Therefore, an interval graph can not contain an induced graph with asteroidal triple. □

Indeed the necessary conditions that we have discussed are also sufficient. The following theorem by Lekkerkerker and Boland that was proved in 1962, shows the necessary and sufficient conditions for interval graphs, in terms of forbidden subgraphs.

Theorem 2.2.10. *(Lekkerkerker-Boland 1962 [42]) A graph G is an interval graph if and only if it does not induce a cycle of length greater than three, and it does not contain an asteroidal triple.*

Here we introduce another property of interval graphs.

Theorem 2.2.11. *(Ghouila-Houri 1962 [25]) The complement of an interval graph has a transitive orientation.*

Proof. Suppose G is an interval graph and let $\{I_v\}_{v\in V(G)}$ be an interval representation for G. Note that two vertices u and v are adjacent in $\bar{G}$ only if u, v are not adjacent in G, i.e., their corresponding intervals I_u and I_v do not intersect. In this case one of these intervals, let's say I_u, lies entirely to left of the other interval. Using this fact we can define the direction from u to v for the uv edge in $\bar{G}$, if and only if I_u is entirely at the left of I_v. To show that this orientation is transitive, suppose $uv \in E(\bar{G})$ is directed from u to v, and $vw \in E(\bar{G})$ is directed from v to w. Therefore in the interval representation of G, the interval I_u lies entirely at the left of I_v and I_v lies entirely at the left of I_w. This implies I_u lies entirely at the left of I_w and the direction for $uw \in E(\bar{G})$ would be from u to v. □

Note that the above property of interval graphs is necessary, but not sufficient. Let G be a 4-cycle. The complement of G is two independent edges. Obviously $\bar{C}_4$ has a transitive orientation, but we know from above that it is not an interval graph.

The following theorem states that the necessary conditions for interval graphs we have already seen are actually sufficient.

Theorem 2.2.12. *(Gilmore-Hoffman 1964 [26])A graph G is an interval graph if and only if G is chordal and the complement of G is a comparability graph.*

Another characterization of interval graphs is based on their maximal cliques. First note that interval graphs are chordal; and we already showed in Theorem 2.2.4 a chordal graph with n vertices has at most n maximal cliques. Assume G is an interval graph with n vertices. The maximal cliques $M_1, \cdots M_t$, $(t \leq n)$ of G can be ordered such that if $v \in M_i \cap M_j$ then v is also a vertex in all M_k for $k = i, \cdots j$.

Indeed each maximal clique M_i can be represented by a point d_i such that all of the corresponding intervals of the the vertices of this maximal cliques have d_i in common. If the the maximal cliques that include a vertex v are not consecutive then there must be some $i < k < j$ such that $v \in M_i, v \in M_j, v \notin M_k$; but this means that $d_i \in I_v, d_k \notin I_v, d_j \in I_v$ which is not possible since $d_i < d_j < d_k$ and I_v is an interval.

Theorem 2.2.13. *(Fulkerson-Gross 1965 [23]) A graph G is an interval graph if and only if maximal cliques of G can be linearly ordered such that, for any vertex $v \in V(G)$ the maximal cliques that contain v appear consecutively.*

Recognition of Interval Graphs

Given a graph G we may need to decide if G is an interval graph. This problem is known as the recognition problem for interval graphs. There are several different linear time (in terms of the number of edges and vertices of the input graph) algorithms for this. Here we briefly mention some of the main ones.

The first interval graph recognition algorithm, by Booth and Leuker in 1967 [3], used Theorem 2.2.13. Their algorithm is based on a data structure called the PQ-tree. A *PQ-tree* is a rooted, labelled tree, in which the leaf nodes represent the endpoints of the intervals, and the internal nodes are labelled P or Q, reflecting different flexibilities of the represented sets of intervals.

The PQ-tree structure is useful in other contexts, but it is complicated. Therefore there were further attempts to find more practical algorithms for interval graph recognition.

Almost twelve years after Booth and Leuker algorithm, Korte and Möhring in 1989 developed a simpler version of PQ-tree, (MPQ-tree) [41]. This algorithm is also based on the property of interval graphs stated in Theorem 2.2.13.

In the previous section we discussed LBFS algorithm for recognizing chordal graphs. Habib et al. in 2000 showed in their paper [30] how to take advantage of LBFS method to design another linear time recognition algorithm for interval graphs. Most recently, in 2009 Corneil et al. have applied LBFS multiple times to recognize interval graphs [8].

2.2.4 Cographs

A *cograph* G is a graph that can be generated from the single vertex graph (K_1) by repeated complementation and disjoint union. In other words if G and H are cographs then so are $\bar{G}$ and $G \cup H$. All cographs are obtained in this way.

It is easy to see that P_4, which is a path consisting of four vertices, is not a cograph. Therefore a cograph does not contain a P_4 as an induced subgraph. Indeed we have the following fact.

Theorem 2.2.14. *[9] A graph is a cograph if and only if it does not contain an induced P_4.*

Here we show that cographs are perfect graphs, by showing that neither a cograph G nor its complement induces an odd cycle. A cograph G cannot induce an odd cycle of length greater than five, because three consecutive edges of the induced odd cycle in G would induce a P_4. On the other hand if v_1, v_2, v_3 and v_4 are consecutive vertices of the induced odd cycle in $\bar{G}$ then in G we have $P_4 = v_3v_1v_4v_2$. Therefore Theorem 2.1.5 implies that G is perfect.

As cographs have forbidden subgraph characterization, they are recognazable in polynomial time. However, cographs may be recognized in linear time, using modular decomposition [9], or partition refinement [31].

2.2.5 Split Graphs

A graph G is a *split* graph if the set of vertices $V(G)$ can be partitioned to two disjoint sets I and Q such that I is independent set and the induced graph on Q is a clique.

The complement of a split graph is also a split graph. Suppose G is a split graph and $V(G) = I \cup Q$, such that I is the independent set, and Q induces a complete graph. Then in $\bar{G}$ the set I induces a clique and Q is an independent set.

Theorem 2.2.15. *[27] A graph G is a split graph if and only $\bar{G}$ is a split graph.*

It is easy to see that if a graph G induces a C_4 or $\bar{C}_4$ or C_5, then G cannot be partitioned to a clique and an independent set, hence is not a split graph. These graphs are the only forbidden subgraphs for a split graph; and if a graph does not induce any of these graphs then it is a split graph.

Theorem 2.2.16. *(Földes-Hammer 1977 [32]) A graph G is a split graph if and only if it does not contain any of the graphs $C_4, \bar{C}_4, C_5$ as an induced subgraph.*

Every split graph G is a chordal graph. This is so, because it does not induce a C_4 or C_5, and moreover if G induce a cycle C_n for $n > 5$ with the set of vertices $\{v_1, \cdots, v_5, \cdots, v_n\}$ then the subgraph induces on $\{v_1, v_2, v_4, v_5\}$ is a $\bar{C}_4$, but Theorem 2.2.16 implies that G does not induce a $\bar{C}_4$.

As split graphs also have finite forbidden subgraph characterization, they are recognizable in polynomial time. However using the degree sequence, split graphs can be recognized linearly [33, 34]. This also follows from the next theorem.

Theorem 2.2.17. *[32] The input graph G is a split graph if and only if both G and $\bar{G}$ are chordal.*

Corollary 2.2.4. *A chordal graph is a split graph if and only if it does not contain an induced $\bar{C}_4$.*

Chapter 3

A Survey Of Matrix Partition

3.1 Introduction

The matrix partition problem was introduced in 1999 in a paper by Feder, Hell, Klein, Motwani [18]. The problem seeks to determine if the vertices of an input graph have a partition into certain parts with adjacencies between parts being restricted by a fixed matrix. The matrix has three different entries, representing full adjacencies, arbitrary adjacencies, or no adjacencies for the corresponding parts, respectively.

The problem can be defined for digraphs but we will only discuss the special case of undirected graphs. So we will always assume that M is a symmetric matrix.

The following is the formal definition of the matrix partition problem.
Let M be a fixed (symmetric) $m \times m$ matrix with entries $\{0, 1, *\}$. For a given graph G, an *M-partition* of G is a partition of vertices of G to parts $P_1, \cdots, P_m$ such that if $M_{ij} = 0$ then each vertex in P_i is non-adjacent to each vertex in P_j, if $M_{ij} = 1$ then each vertex in P_i is adjacent to all vertices in P_j, and if $M_{ij} = *$ then there is no restriction for adjacencies between vertices in P_i and P_j (vertices in P_i can be arbitrarily adjacent to vertices in its part).

Note that when $i = j$ the definition is taken to describe the adjacencies of a vertex to all other vertices of its part, i.e., the entries on the main diagonal of M give the restrictions on the individual parts. If $M_{ii} = 0$ then P_i is an independent set, if $M_{ii} = 1$ then P_i is a clique, and if $M_{ii} = *$ then P_i is a set of vertices with arbitrary connections (each vertex in P_i is adjacent to some, possibly all or none, of the other vertices in P_i).

For a visual concept of a partition problem we use symbolic figures for the structure of partitioned graphs. In this visualization, an empty circle shows an independent set (equivalent to a zero on the main diagonal of the matrix), a grey (light grey) circle shows a part with arbitrary connections (equivalent to an asterisk on the main diagonal of the matrix) and a black (dark grey) circle shows a clique part (equivalent to a one on the main diagonal of the matrix). Also, two parts are joined with one line, if they are arbitrarily adjacent (equivalent to an asterisk off-diagonal entry of the matrix); two parts are joined with double lines, if they are fully connected (equivalent to a one as an entry off the main diagonal).

Here are three examples to illustrate the concept of the matrix partition problem.

First Example: Let M for the matrix partition problem be:

$$M = \begin{pmatrix} 0 & * \\ * & 1 \end{pmatrix}$$

The following is the symbolic figure of M.

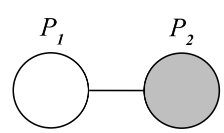


Figure 3.1: The structure of split graphs.

This M-partition problem corresponds to a partition of a graph into an independent set (as $M_{1,1} = 0$) and a clique (as $M_{2,2} = 1$). There is no restriction for the edges between these two parts (as $M_{1,2} = M_{2,1} = *$). Indeed a graph G is partitionable according to M, if and only if G is a split graph. Suppose the input graph G is the graph in Figure 3.2(a) (known as the *bull* graph), you can see M-partition of G in Figure 3.2(b).

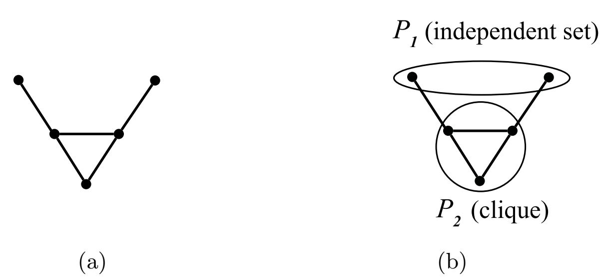


Figure 3.2: The M-partition of the bull graph.

Second Example: Let the matrix for this example be:

$$M' = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & * \\ 0 & * & 1 \end{pmatrix}$$

The following is the symbolic figure of M'.

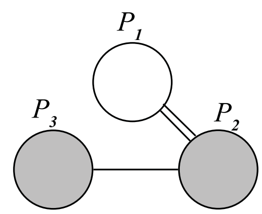


Figure 3.3: The structure of M'-partitioned graphs.

For this partition problem, the first part is an independent set, the second and third parts are cliques. The first part has all the possible edges to the second part, and is independent from the third part. There is no restriction for the edges between the second and third parts. Suppose G is the graph in Figure 3.4(a) (know as the *house* graph); you can see M'-partition of G in Figure 3.4(b).

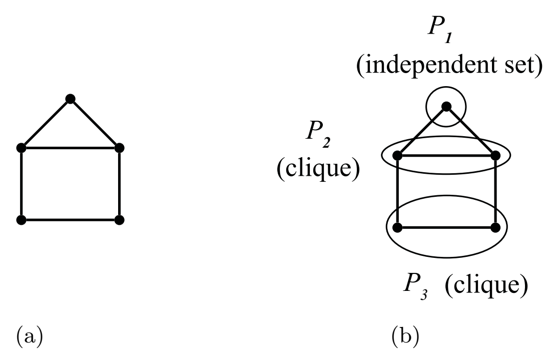


Figure 3.4: The M'-partition of the house graph.

Third Example: Suppose the matrix for this example is

$$M'' = \begin{pmatrix} 0 & 1 & * & 0 \\ 1 & 0 & * & 1 \\ * & * & 1 & 1 \\ 0 & 1 & 1 & * \end{pmatrix}$$

The following is the symbolic figure of M''.

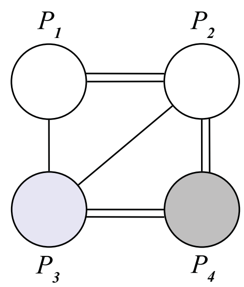


Figure 3.5: The structure of M''-partitioned graphs.

Note that there is an asterisk entry on the main diagonal of M'', so the partition problem is trivial. Putting all of the vertices of the given graph G in the fourth part is an M-partition of G. Assume the input graph G is the graph which is in Figure 3.6(a) (known as the the *diamond* graph); you can see an M''-partition of G in Figure 3.6(b).

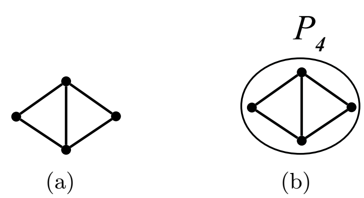


Figure 3.6: The M''-partition of the diamond graph.

Here we discuss how the matrix partition is related to two previously discussed problems: the k-colouring and the homomorphism problems.

Let M be the $k \times k$ matrix with all diagonal entries equal to 0 and all off-diagonal entries equal to $*$. Then the M-partition problem seeks k independent sets in the input graph G, such that edges between each pair of independent sets are arbitrary. Thus a graph is M-partitionable if and only if it is k-colourable. We can say that k-colouring problem is a special case of the partition problem such that M is obtained from the adjacency matrix of K_k by replacing 1's by $*$'s.

For another case, consider a graph H with vertices: $v_1, \cdots, v_n$. Suppose M is the adjacency matrix of H with the 1's again replaced by $*$'s. An M-partition of an input graph G, is indeed a homomorphism of G to H. Because if u and v are adjacent in G, and u maps to the part P_i and v maps to the part P_j in an M-partition of G, then $M_{i,j}$ cannot be zero. Therefore $M_{i,j} = *$ implies that in the adjacency matrix of H, the entry corresponding to $v_i v_j$ is one; so v_i the image of u and v_j the image of v are adjacent in H.

Since both the colouring and the homomorphism problems are special cases of the matrix partition problem, the matrix partitions are sometimes referred to as a generalized colourings.

In the following we introduce some well-known partition problems, and their interpretation with the matrix partition terminology; and the connection of these problems to the study of perfect graphs.

A *clique cutset* of a connected graph G is a complete subgraph C whose removal disconnects G. There is a clique cutset in G if and only if vertices in G can be partitioned in three non-empty sets A, B, C such that C is a clique and A, B are independent of each other. With the matrix partition notation, a graph has a clique cutset if it is partitionable according to the matrix M_1 shown below, with the additional constraint that all parts are non-empty.

$$M_1 = \begin{pmatrix} * & 0 & * \\ 0 & * & * \\ * & * & 1 \end{pmatrix}$$

The constraint of non-empty parts makes this problem non-trivial, although there are asterisk entries on the main diagonal of M. Nevertheless finding clique cutsets is polynomial [50, 54, 55]. Clique cutsets were originally studied because of their connection to chordal graphs. Any chordal graph has a clique cutset, and clique cutsets can decompose chordal graphs completely. Specifically, suppose chordal graph G has a clique cutset C. If the connected components of $G \backslash C$ are $G_1, \cdots G_t$, then each C union G_i, for $i = 1, \cdots, t$ which is a chordal graph has also a clique cutset, we can continue doing this procedure until there is just one vertex remains in each component [50]. The symbolic figure of M_1 is shown in Figure 3.7.

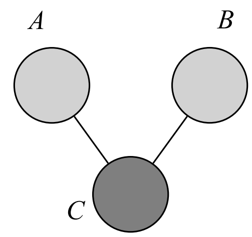


Figure 3.7: A clique cutset C in M_1-partitioned graphs.

Next consider the stable cutset problem.

A *stable cutset* of a connected graph G is an independent set C whose removal disconnects G. As described above for the clique cutset problem, the input graph G has a stable cutset if and only if vertices in G can be partitioned in three non-empty sets A, B, C such that C is a stable (independent) set and A, B are independent of each other. We can view this problem as a matrix partition according to the following matrix M_2 with the additional constraints of non-empty parts.

$$M_2 = \begin{pmatrix} * & 0 & * \\ 0 & * & * \\ * & * & 0 \end{pmatrix}$$

The matrix M_2 is similar to the matrix M_1, but the entry at the third row and column is zero. The stable cutset problem is NP-complete [38]. Stable cutsets have a close relation with perfect graphs. Note a *minimal imperfect* graph is a graph which is not perfect, but every induced subgraph of it is perfect. Minimal imperfect graphs, except the odd cycles cannot contain a stable cutset [51].

The symbolic figure of M_2 is shown in Figure 3.8.

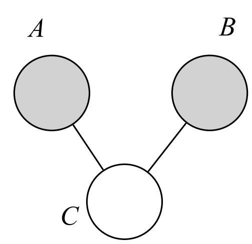


Figure 3.8: A stable cutset C in M_2-partitioned graphs.

Next, we introduce another known problem which can be interpreted as a matrix partition problem. A *homogeneous set* in a graph G is a set C such that each vertex outside of C is adjacent to either all or to none of the vertices in C. In other words, vertices in C have the same neighbourhood outside of C. So for this problem, we wish to partition the vertices of G into three sets A, B and C, such that A, C are independent of each other, and B, C are completely adjacent. Note that a single vertex is a homogeneous set; so is the set of all vertices of G (both A and B are empty). Therefore we consider that $A \cup B$ is not empty and C contains more than one vertex. So this problem is indeed a matrix partition problem according to the following matrix M_3, with the additional constraints that the first and second parts are not both empty, and there are at least two vertices in the third part.

$$M_3 = \begin{pmatrix} * & * & 0 \\ * & * & 1 \\ 0 & 1 & * \end{pmatrix}$$

Homogeneous sets can facilitate the recognition of comparability graphs (and other similar classes of graphs) [44, 10]. It is again the case that minimal imperfect sets cannot have a homogeneous set. This fact was used by Lovász to prove the weak perfect graph conjecture [43]. Homogeneous sets can be found efficiently in linear time [44]. The following is the symbolic figure of a graph with a homogeneous set C.

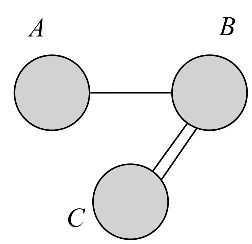


Figure 3.9: A homogeneous set C in M_3-partitioned graphs.

Another problem is the skew cutset problem. A *skew cutset* of a connected graph G is a pair of disjoint non-empty sets of vertices B and D in G such that $G \backslash B \cup D$ is not connected; also all the edges between the vertices in B and D are present (the skew property). This problem is indeed a partition problem. We wish to partition G into four non-empty sets A, B, C and D such that: A, C are independent of each other and B, D are completely adjacent. A graph has a skew cutset if it is partitionable according to the matrix M_4, with the additional constraint of having non-empty parts.

$$M_4 = \begin{pmatrix} * & * & 0 & * \\ * & * & * & 1 \\ 0 & * & * & * \\ * & 1 & * & * \end{pmatrix}$$

The main diagonal entries of M_4 are all asterisks. Skew cutset partition plays an important role in the proof of the strong perfect graph conjecture [6]. Finding skew cutsets has a long history. Chvatal [7] asked if it could be done in polynomial time. Feder et al [18] proved it can be done in sub-exponential time using the list version of the problem, and then Reed et al [11] found a polynomial time algorithm also based on the list version.

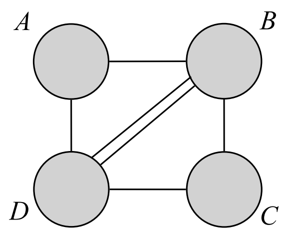


Figure 3.10: A skew cutset in M_4-partitionable graphs.

The list version of the M-partition problem asks whether a given input graph G, with a list of admissible parts for each vertex, has an M-partition. (So in an M-partitioned of G each vertex must be placed in a part in its list.) Many of the variants discussed above can be modelled as list M-partitions. These include non-empty parts, or parts with at least two elements (as we have in the homogeneous set problem).

For example the M-partition of G with the constraint that one part, say P_1, contains at least two elements can be modelled as a list partition problem: suppose there are m parts, i.e., M is of size $m \times m$, and the input graph G has n vertices. We consider $\binom{n}{2}$ instances of list M-partition for G. Each one is corresponding to select two vertices and only assign P_1 to their lists, and other vertices have all of the parts in their lists. Then G is M-partitionable with the additional constraint for P_1, if and only if G is list M-partitionable for any of the instances.

But here, for the rest of this study, we suppose there is no restriction for having non-empty parts, and there are no constraints for vertices to map to certain parts, i.e, no lists for the vertices, or for each $v \in V(G)$ we have $\ell(v) = \{P_1, \cdots, P_m\}$.

From now on we assume that no entry on the main diagonal of M is an asterisk. So we assume that each part is either an independent set or a clique, i.e., all diagonal entries of the matrix are either zero or one. It is always possible to arrange the matrix in a way that on the main diagonal, first all zeros appear and then all ones. Suppose there are k independent sets and ℓ cliques in the matrix partition problem. For such a problem, we can consider the matrix M consisting of two diagonal block matrices A and B; where A is a k by k zero diagonal matrix (the $k \times k$ upper left submatrix of M), and B is an ℓ by ℓ one diagonal matrix (the $\ell \times \ell$ lower right submatrix of M).

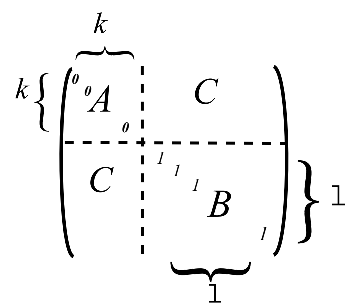


Figure 3.11: The assumed form of a matrix.

3.2 First Order Definability and Minimal Obstructions

In logic, *first order formulas* handle atomic variables, relations, quantifiers over variables, and connectives such as negation, conjunction, and disjunction. The term *atomic* refers to a variable or a declarative sentence (proposition) which is either true or false and which cannot be broken down into other simpler components. First order formulas, naturally define graph classes and we shall sometimes refer to this classification.

Let $x_1, x_2 \cdots$ be variables standing for vertices of the graph G, and let E be the binary relation standing for the edges of G. Then a first order formula is anything that uses the symbols $x_1, x_2 \cdots$, the symbol E, the universal and existential quantifiers, and the usual propositional connectives.

For example, consider the following formulas

$$F(G) = \neg\{\ \exists x_1 \exists x_2 \exists x_3\ [\ (x_1x_2 \in E) \wedge (x_1x_3 \in E) \wedge (x_2x_3 \in E)\ ]\ \}$$

$$F'(G) = \neg\{\ \exists x_1 \exists x_2 \exists x_3\ [\ (x_1x_2 \in E) \wedge (x_2x_3 \in E) \wedge \neg(x_1x_3 \in E)\ ]\ \}$$

The graphs G that satisfy the first formula are precisely the graphs without an induced triangle; and those that satisfy the second formula, are precisely the graphs that do not induce a path with three vertices.

We use the abbreviation *FOD* to refer to the set of problems for which the positive instances can be described by a first order definable formula. In particular, for M-partition problems, we use FOD to denote the set of matrices M for which there exists a first order formula F such that G is M-partitionable if and only if G satisfies F.
For example, for the 1×1 matrix $M = (0)$, a graph G is M-partitionable if and only if G has no edges, and having no edges is expressible by an obvious first order formula, so M is in FOD.

Next we define another concept which is useful for our research approach.

In the matrix partition terminology, an *obstruction* refers to a graph which is not partitionable according to a given matrix M, but any proper induced subgraph of it is M-partitionable. For example consider M as the following matrix, which seeks two independent sets in a graph that are arbitrarily connected to each other.

$$M = \begin{pmatrix} 0 & * \\ * & 0 \end{pmatrix}$$

Any graph which is partitionable according to this matrix is bipartite. Any odd cycle is a minimal obstruction for this M-partition problem. This is so since odd cycles are not bipartite and each of their proper induced subgraph is bipartite. As you can see the number of minimal obstructions for the partition problem according to the above matrix is infinite, i.e., there are infinitely many minimal obstructions for this problem, i.e., the number of minimal obstructions is infinite. If we consider this partition problem (according to the above matrix M) for chordal graphs, then the only minimal obstruction is a triangle. Therefore this M-partition problem for chordal graphs has only finitely many minimal obstructions. Using an abbreviation, we refer to the set of problems (matrices) with finitely many minimal obstructions as *FMMO*.

It is easy to see that if there are only finitely many minimal obstructions for a problem, then there is a first order formula for it. Indeed, we can write the formula as a negation of a disjunction of conjunctive clauses, each corresponding to the absence of a forbidden induced subgraph, in a way similar to the formulas $F(G)$ and $F'(G)$ described earlier. For example assume C_3 and P_3 are forbidden subgraphs for a class of graphs. Then any graph G that satisfies $F''(G) = F(G) \wedge F(G')$ is a member of this class. (Note that F, F' are the functions we defined earlier.)

Note that checking if a graph G with n vertices contains an induced subgraph H with k vertices can be done in polynomial time (note that H is a fixed graph, so k is constant). One brute-force algorithm is to consider all subgraphs with k vertices and check if any of them is H. There are $\binom{n}{k}$ such subgraphs, therefore checking if G contains H is of $O(n^k)$. Thus if there are finitely many minimal obstructions for a problem then it is polynomial. We refer to the problems with polynomial time solution as class of $\mathcal{P}$. Therefore if a problem is in $FMMO$ then it is in $\mathcal{P}$ as well.

Furthermore, it has recently been shown [13] that for many classes of graphs with not too many edges (for example graphs with bounded degree or graphs that can be drawn on a given surface with at most constant number of crossings per edge) problems in FOD can be solved in linear time.

Returning to our example, as we stated in Chapter 1, deciding if a graph is bipartite is polynomial solvable. So there are polynomial time solvable problems with infinitely many minimal obstructions and hence FMMO is not equal to $\mathcal{P}$.

Summarizing the inclusions between the three classes of problems we have the following theorem.

Theorem 3.2.1. *[48, 28] FMMO $\subseteq$ FOD $\subset$ $\mathcal{P}$*

□

Whether the first set is proper subset of the second set or not is still an open question. However, it follows from a general result [48] that FOD=FMMO for homomorphism problems, i.e., for the M-partition problems where M has no 1's.

Theorem 3.2.2. *[48] If M has no 1's, then the M-partitionable graphs are first order definable if and only if they have finitely many minimal obstructions.*

3.3 Matrix Partition Results

In this section we review the current results for the matrix partition problem, considering the complexity and finiteness versus infiniteness of the set of minimal obstructions.

3.3.1 General Graphs

In this section we state the results of the matrix partition problem that hold for all graphs in general.

Note that the *complement of a matrix* M, denoted by $\bar{M}$, is M whose 1's are replaced by 0, and 0's are replaced by 1's. The asterisk entries of M are kept fixed in $\bar{M}$. We rearrange the parts so that we first have 0's and then 1's on the main diagonal. For an example consider M for split graphs.

$$M = \begin{pmatrix} 0 & * \\ * & 1 \end{pmatrix} \qquad \bar{M} = \begin{pmatrix} 0 & * \\ * & 1 \end{pmatrix}$$

Note that in this case $M = \bar{M}$. We have the following fact.

Theorem 3.3.1. *A graph G is M-partitionable if and only if $\bar{G}$ is $\bar{M}$-partitionable.*

We do not have a known classification for the complexity of the matrix partition for general graphs. However there are some results for small matrices (when M is at most 4×4), that classify the complexity of M-partition.

Theorem 3.3.2. *If M is a 2×2 matrix, then the M-partition problem is polynomial time solvable.*

Proof. The M-partition problem when M is a 2×2 matrix, can be considered as a 2-SAT problem. For the input graph G, the boolean variables of the 2-SAT instance are x_v for each $v \in V(G)$, such that $x_v = 1$ if v maps to the first part, and $x_v = 0$ if v maps to the second part. The clauses in the 2-SAT formula are as follows: for vertices u and v such that $uv \in E(G)$, $(x_v \vee x_u)$ if $M_{2,2} = 0$, $(\bar{x}_v \vee \bar{x}_u)$ if $M_{1,1} = 0$, $(\bar{x}_v \vee x_u) \wedge (x_v \vee \bar{x}_u)$ if $M_{1,2} = M_{2,1} = 0$ and for non-adjacent vertices u and v we add, $(x_v \vee x_u)$ if $M_{2,2} = 1$, $(\bar{x}_v \vee \bar{x}_u)$ if $M_{1,1} = 1$, $(\bar{x}_v \vee x_u) \wedge (x_v \vee \bar{x}_u)$ if $M_{1,2} = M_{2,1} = 1$. This instance of 2-SAT is satisfiable if and only if G is M-partitionable. Because if G is M-partitionable, the partition gives the truth values of the variables. On the other hand, assignments of variables in the satisfiable 2-SAT instance gives the partition. Solving 2-SAT is polynomial in number of variables and clauses, and for this problem the number of variables and clauses is polynomial in the number of vertices in G. □

Theorem 3.3.3. *[19] Let M be a matrix of size 3×3 or 4×4, then the M-partition problem is NP-complete if M contains the matrix corresponding to 3-colouring or its complement; otherwise, it is polynomial time solvable.*

Recall that M, the matrix for 3-colouring and its complement are as follows:

$$M = \begin{pmatrix} 0 & * & * \\ * & 0 & * \\ * & * & 0 \end{pmatrix} \quad \bar{M} = \begin{pmatrix} 1 & * & * \\ * & 1 & * \\ * & * & 1 \end{pmatrix}$$

The following corollary summarizes the complexity results that we state in the last two theorems.

Corollary 3.3.1. *[18] Let M be a matrix of size at most 4×4. Then the M-partition problem is NP-complete if M contains a submatrix for 3-colouring or complement of 3-colouring; otherwise it is polynomial.*

The following theorem is a general result on the complexity of the matrix partition for general graphs. Before stating the theorem, note that a function is *quasi-polynomial* if it is bounded by $n^{c\log^t n} = 2^{c\log^{t+1} n}$ for the input n and some positive constant c, t. Note that it is a sub-exponential function. Indeed, it is conjectured that NP-complete problems do not have sub-exponential time algorithms. Therefore if there is a quasi polynomial algorithm for a problem, it is most likely that the problem is not NP-complete.

Theorem 3.3.4. *[19] If an M-partition problem is not NP-complete, then there is a quasi-polynomial algorithm to solve it.*

We do not have a known classification for the finiteness minimal obstructions for the matrix partition problem. However when the size of the matrix is small there exist such classification.

The following theorem gives a class of matrices in FMMO. The proof by Feder and Hell [16] is complex, however there are shorter proofs and we present another short proof here.

Theorem 3.3.5. *[16] If M is a matrix without any $*$ entry, then the M-partition problem has only finitely many minimal obstructions.*

Proof. Suppose M has k zeros and ℓ ones on its main diagonal, and any off-diagonal entry is either a zero or a one. Any minimal obstruction has at most $1 + 2(k + \ell)(\max\{k, \ell\} + 1)$ vertices. This bound on the number of vertices implies there are only finitely many minimal obstructions.

We may assume that $\max(k, \ell) = k$ i.e., that $\ell \leq k$; the other case when $\max(k, \ell) = \ell$ we can use the complement of the graph and the matrix. Suppose there is a minimal obstruction

G with more than $1+2(k+\ell)(k+1)$ vertices. Therefore G has at least $2+2(k+\ell)(k+1)$ vertices and, for any vertex v in G, the graph $G\backslash\{v\}$ is M-partitionable. Consider a vertex v and an M-partition of $G\backslash\{v\}$. There are at least $1+2(k+\ell)(k+1)$ vertices in $G\backslash\{v\}$. As there are $k+\ell$ parts, by the pigeon hole principle, there is a part, say P, which contains at least $1+2(k+1)=2k+3$ vertices. This P is either a clique or an independent set. Each of these two cases will be considered separately. All vertices in one part, specifically in P, are homogeneous in $G\backslash\{v\}$: because there is no $*$ in M, all the vertices in one part have the same neighbours in other parts. So, there are at least $2k+3$ homogeneous vertices in $G\backslash\{v\}$, which implies at least $k+2$ homogeneous vertices in G. In fact, any vertices in P can be adjacent or non-adjacent to v, the pigeon hole principle, at least $k+2$ of the vertices in P are in the same adjacency situation to v. Let's call this set Q which is a subset of P. Consider some $w \in Q$ and recall that $G\backslash\{w\}$ is also M-partition.

First, consider P is an independent set, so Q is also an independent set. Since there are $k+1$ independent vertices in $Q\backslash\{w\}$, there are at least $k+1$ independent vertices in $G\backslash\{w\}$. As there are ℓ clique parts in the partitioned graph, and $\ell \leq k$, and no two independent vertices can be mapped to one clique part, at least one of the vertices of $Q\backslash\{w\}$, say w', is mapped to an independent set. This vertex is similar to w and there is no edge between them, so w can be mapped to the same part as w' giving a partition for G, which is impossible.

If P is a clique, then $Q\backslash\{w\}$ with $k+1$ vertices is also a clique. So, as above, in a partition of $G\backslash\{w\}$, there is at least one vertex of $Q\backslash\{w\}$ that must be mapped to a clique; so w can be mapped to this clique to obtain a partition for a G which is again, a contradiction.

□

This theorem implies that such M-partition problem admits finitely many minimal obstructions in any class of graphs.

A more general class of matrices has been defined by Feder, Hell and Xie [21] that we use in the statement of the next two theorems. A *friendly* matrix is a matrix without any $*$ entry in A and B, according to the Figure 3.11. A matrix is *unfriendly* if it is not friendly, i.e., it has some $*$ entry in A or B. We have the following fact.

Theorem 3.3.6. *[21] If M is an unfriendly matrix, there are infinitely many minimal obstructions for M-partition.*

For small friendly matrices we have the converse:

Theorem 3.3.7. *[21] If M is a friendly matrix of size $k \times k$ with $k \leq 5$, then there are just finitely many minimal obstructions for M-partition.*

This theorem cannot be generalized for bigger matrices. For instance, consider the following matrix M:

$$M = \begin{pmatrix} 0 & 1 & 0 & * & 0 & 0 \\ 1 & 0 & 1 & * & 0 & 0 \\ 0 & 1 & 0 & 0 & * & * \\ * & * & 0 & 1 & 0 & 1 \\ 0 & 0 & * & 0 & 1 & 0 \\ 0 & 0 & * & 1 & 0 & 1 \end{pmatrix}$$

There are infinitely many minimal obstructions for this M-partition problem [21].

3.3.2 Perfect Graphs

Many results of the matrix partition have been found for restricted classes of graphs. In this section we focus on perfect graphs.

First note that Theorem 3.3.1 can be applied to perfect graphs. This is so, because the weak perfect graph theorem, Theorem 2.1.4, implies that the complement of a perfect graph is also perfect.

In this section we focus more on minimal obstructions. Recall that when M is in FMMO, then the M-partition problem is polynomial.

Referring to Figure 3.11, we use $S(A)$ to denote the set of off-diagonal entries of A, and $S(B)$ to denote the set of off-diagonal entries of B, and $S(C)$ to denote the set of all entries in C. We use $\mathcal{M}(X, Y, Z)$ to denote the set of all matrices M such that $S(A) \subseteq X, S(B) \subseteq Y, S(C) \subseteq Z$. With this notation class of friendly matrices can be written as $\mathcal{M}(\{0,1\}, \{0,1\}, \{0,1,*\})$ When each of $S(A), S(B)$ or $S(C)$ is a singleton (a set with just one member), we refer to it just by its member; thus $\mathcal{M}(a, b, c)$ is the set of matrices with $S(A) = a$, $S(B) = b$ and $S(C) = c$. We assume $a \neq 0$ and $b \neq 1$; because if $a = 0$ then A is just a matrix with zeros, and we may assume that k (the size of A) is just one. A similar argument applies for the case of $b = 1$.

Theorem 3.3.8. *[15] If $c \neq *$, and $M \in \mathcal{M}(a, b, c)$ then each minimal perfect obstruction for M-partition has at most $(k + 1)(\ell + 1)$ vertices.*

In [15], the authors also consider a generalization to $\mathcal{M}(X, Y, c)$ where X and Y are sets.

So far, we have focused on results where $c \neq *$. The following theorem, assumes $c = *$.

Theorem 3.3.9. *[15] Let $M \in \mathcal{M}(a, b, *)$. Then there are finitely many minimal perfect obstructions for the M-partition problem.*

For the case where $c = *$, there is also a generalization for $\mathcal{M}(X, Y, *)$ with X and Y being sets [15].

In all these cases, the bound for the number of vertices in each minimal perfect obstruction is $1 + 2((1 + \max(k, \ell))^2)^{2k\ell+1}$. This bound is exponential of size of the M, but as M is fixed it implies a polynomial time algorithm for the M-partition problem considering the obstructions.

For the next theorem we define normal matrices, just to have a cleaner statement. We shall say that a subset of $\{0, 1, *\}$ is *normal* if it does not contain both an $*$ and another element. Thus every normal set is either $\{*\}$ or a subset of $\{0, 1\}$. A *normal matrix* is a matrix such that each of $S(A), S(B)$ and $S(C)$ is normal (not necessarily with the same choice of $\{*\}$ or $\{0, 1\}$.

Theorem 3.3.10. *[15] If M is a normal matrix, then there are only finitely many minimal perfect obstructions for the M-partition problem.*

3.3.3 Chordal Graphs

In this section we focus on chordal graphs. For chordal graphs, also, there is no known classification of the complexity of M-partition, or the finiteness of the number of minimal obstructions. Chordal graphs have a perfect elimination ordering, which is often helpful in designing algorithms. On the other hand, the complements of chordal graphs are not necessarily chordal, so we cannot use the Theorem 3.3.1 for this class of graphs.

In this sections we first discuss some of the results for complexity and then obstructions of the matrix partition problem for chordal graphs.

In the following, Theorem 3.3.11 says that partitioning a chordal graph into some independent sets is polynomial; and Theorem 3.3.12 says that partitioning a chordal graph into some cliques is polynomial. They also give the only minimal obstructions for each case.

Theorem 3.3.11. *[34] Let M be a matrix of size $k \times k$ such that all entries on the main diagonal are 1's, and all off-diagonal entries are $*$. Then G has an M-partition if and only if it has no induced K_{k+1}.*

This partition problem is indeed k-colouring of input chordal graph G. Although k-colouring for general graphs is NP-complete, it is linear (of the size of vertices and edges) for chordal graphs.

Theorem 3.3.12. *[34] Let M be a matrix of size $\ell \times \ell$ such that all entries on the main diagonal are 0's, and all off-diagonal entries are $*$. Then G has an M-partition if and only if it has no induced $\bar{K}_{\ell+1}$.*

This partition problem is covering the input graph G by ℓ cliques, which is an NP-complete problem for general graphs, and linear time for chordal graphs.

Theorem 3.3.13 extends Theorems 3.3.11 and 3.3.12.

Theorem 3.3.13. *[34] Let M be a matrix with all off-diagonal entries being $*$. The M-partition problem for chordal graphs is polynomial, and the only minimal chordal obstruction is $(\ell+1)K_{k+1}$ (disjoint union of $\ell+1$ complete graphs of size $k+1$).*

This theorem states that $\mathcal{M}(*,*,*)$ is a family of matrices that the matrix partition problem is polynomial for choral graphs (linear indeed). The linear algorithm for this partition problem is using the perfect elimination ordering of the input chordal graph. This problem has first been studied by Brandstädt [4]. This problem is NP-complete for general class of graphs, unless $k \leq 2$ and $\ell \leq 2$.

The next theorem is for a class of matrices called crossed. A matrix M is called *crossed* if each non-$*$ entry in its block C, belongs to a row or a column of non-$*$ entries.

Theorem 3.3.14. *[20] If M is a crossed matrix then the M-partition problem for chordal graphs is polynomial.*

It is proved [20] that for an input chordal graph G with n vertices and crossed matrix M, the M-partition problem is solvable in time $n^{k\ell}$.

Note that as class of chordal graphs is a subset of the class of perfect graphs, then Theorem 3.3.10 is also valid for chordal graphs.

3.3.4 Cographs

Cographs are the largest class of graphs that the complexity of the matrix partition is completely solved. Indeed the matrix partition problem is always polynomial for co-graphs.

All forms of matrix partition problem are solvable in polynomial time and there are just finitely many minimal obstructions.

Theorem 3.3.15. *[17] Suppose M is a matrix of size $m \times m$. Any minimal cograph obstruction for the M-partition problem has at most ${}^{8^m}/_{\sqrt{m}}$ vertices; there exist matrices M for which some minimal cograph obstructions have ${}^{m^2}/_4$ vertices.*

One special case of matrix partition problem is when all the off-diagonal entries of M are $*$, i.e., the family of matrices in $\mathcal{M}(*, *, *)$. We discussed this problem in the chordal section. Here we mention that the minimal cograph obstruction and minimal chordal obstruction are the same.

Theorem 3.3.16. *[17] If M is a matrix with all off-diagonal entries equal to $*$, then each minimal cograph obstruction has exactly $(k+1)(\ell+1)$ vertices.*

Chapter 4

New Results

4.1 Our Approach

There are three well-known approaches to classify matrix partition problems. These approaches are, a classification according to their first order definability, a classification according to their computational complexity, and a classification according to the finiteness or infiniteness of the set of minimal obstructions. Of these, we mostly deal with the classification according to the finiteness or infiniteness of the number of minimal obstructions. We focus on matrices of size 3×3 with the same entry on the main diagonal, i.e., either matrices with just zeros on the main diagonal, or matrices with just ones on the main diagonal. For cases that were previously open, we have classified which have finitely and which have infinitely many minimal chordal obstructions. These results appear as Theorems 4.3.1, 4.3.2, 4.3.3, and 4.3.4. Each of these cases are polynomial time solvable for the class of chordal graphs [15].

4.2 Useful Tools

In this section we introduce and review some useful techniques for our research to classify the matrix partition problem. They are implicit in the literature but we state them explicitly for later use.

4.2.1 Matrices With Two Kinds of Entries

In this section we prove that any matrix M with only two kinds of entries is in FMMO. One case is when the entries of M are $\{0, 1\}$. This case M is a matrix without asterisk entries, and we can apply Theorem 3.3.5. The other cases are when the entries of M are $\{0, *\}$ or $\{1, *\}$.

Let M be a matrix without 1's. Let M' be a matrix corresponding to M but with all $*$'s replaced by 1's, and let H_M be the graph whose adjacency matrix is M'. We can view an M-partition of G as a homomorphism $f : G \to H_M$. (A vertex of G is assigned to a part, i.e., a vertex of H_M.)

Lemma 4.2.1. *A perfect graph G is M-partitionable if and only if $\omega(G) \leq \omega(H_M)$.*

Proof. Note that, when there are no 1's in M, all entries on the main diagonal of M are zeros, by our general assumption; therefore, all parts are independent sets.

Note that f cannot map two adjacent vertices of G to the same vertex of H_M, as vertices of H_M correspond to independent sets. This implies that in any M-partition of G, a clique of size k in G must be placed in parts corresponding to a clique of size k in H_M; therefore $\omega(G) \leq \omega(H_M)$.

Now it will be shown that when $\omega(G) \leq \omega(H_M)$, then G is M-partitionable. Since G is chordal and hence perfect, it can be partitioned into $\omega(G) \leq \omega(H_M)$ independent sets. The largest clique in H_M has $\omega(H_M)$ vertices, and corresponds to $\omega(H_M)$ independent parts in M with asterisk adjacencies among them. We assign each of the $\omega(G)$ independent sets in G to a exactly one of these independent parts. $\square$

The next theorem follows from the above Lemma.

Theorem 4.2.1. *If M is a matrix without 1's, then the M-partition problem admits only finitely many perfect minimal obstructions.*

Proof. The only minimal obstruction for this partition problem is complete graph with $\omega(H_M)+1$ vertices. Lemma 4.2.1 implies, $K_{\omega(H_M)+1}$ is not M-partitionable, since its clique size is greater than $\omega(H_M)$, but omitting each of its vertices the remaining graph, $K_{\omega(H_M)}$, is M-partitionable. $\square$

This theorem also holds for chordal graphs, with the same minimal obstruction.

Corollary 4.2.1. *If M is a matrix without 0's, then the M-partition problem admits only finitely many perfect minimal obstructions.*

Proof. Here $\bar{M}$ is a matrix without 1's. Theorem 3.3.1 implies G is M-partitionable if and only if $\bar{G}$ is $\bar{M}$-partitionable. $\square$

4.2.2 Domination

Lemma 4.2.2. *Let $x \neq *$ and let M and M' be as the following:*

$$M = \begin{pmatrix} 0 & 0 & * \\ 0 & 0 & y \\ * & y & x \end{pmatrix} \quad \textit{and} \quad M' = \begin{pmatrix} 0 & * \\ * & x \end{pmatrix}.$$

A graph G is M-partitionable if and only if it is M'-partitionable.

Proof. If a graph is M'-partitionable, it is also M-partitionable with an empty set as the second part. If an input graph is M-partitionable, we can move any vertex from the second part and put it in the first part, and still have a proper M-partition. In fact the first and second parts are independent sets, and there is not any edge between them, so together they can be observed as one independent part which is arbitrarily connected to the third part. In this way G is M'-partitioned. $\square$

Lemma 4.2.3. *For any $x \in \{0, 1\}$ and $y \in \{0, 1, *\}$, let*

$$M = \begin{pmatrix} x & y & * \\ y & 1 & 1 \\ * & 1 & 1 \end{pmatrix} \quad \textit{and} \quad M' = \begin{pmatrix} x & * \\ * & 1 \end{pmatrix}.$$

A chordal graph G is M-partition if and only if it is M'-partitionable.

Proof. Here the proof is an analogous argument to the previous case. An M'-partitionable graph is M-partitionable with an empty second part. On the other hand, if G is M-partitionable, any vertex in the second part can move to the third part, because each vertex in the second part is adjacent to all the vertices in the second part and the third part. So the second and third parts, together, form a clique with arbitrary connections to the first part. Thus this is an M'-partition of G. $\square$

4.2.3 Pre-Colouring

In certain cases, it will be useful to consider some vertices of G labelled - or *pre-coloured* - with the names the of parts.

Consider the set $\mathcal{G}$ of all chordal graphs with up to three independent vertices *pre-coloured* with different colours from $1, 2, 3$.

Definition 4.2.1. *Let G be a graph with a set of up to three independent vertices pre-coloured by different colours. Then G is a minimal pre-coloured obstruction for M-partition problem if*

- *there is no M-partition of G in which the pre-coloured vertices map to the corresponding parts,*
- *for any $v \in V(G)$, the graph $G \backslash \{v\}$, with the inherited pre-coloured vertices does admit an M-partition in which the pre-coloured vertices map to the corresponding parts.*

We say that a pre-coloured graph G *contains* a pre-coloured graph G' if G' is an induced subgraph of G and each pre-coloured vertex in G' is also pre-coloured in G, with the same colour. (Thus G' can have some vertices not pre-coloured even if they are pre-coloured in G.) A *pre-coloured obstruction* is one that does not properly contain another pre-coloured obstruction.

Theorem 4.2.2. *Let M be an 3×3 matrix with all diagonal entries equal to 1. Suppose all chordal minimal pre-coloured obstructions to M-partition have at most k vertices. Then all chordal minimal obstructions that contain three independent vertices have at most $3+6(k-3)$ vertices.*

Proof. Let G be a minimal obstruction for the M-partition problem and u, v, w three independent vertices in G. Since all diagonal entries of M are 1, the vertices u, v, w must obtain different colours. There are 6 permutations possible, and each of them must result in a pre-coloured obstruction with at most k vertices, $k-3$ of which are different from u, v, w. Thus by minimality of G, it has at most $3 + 6(k-3)$ vertices. $\square$

4.3 Small Matrices

In this section we study 2×2 and some 3×3 matrices and classify them according to whether or not the number of minimal chordal obstructions is finite. It appears that a complete classification, even for 3×3 matrices, is quite demanding. For 3×3 we will mostly focus on matrices with constant diagonals (either all zeros or all ones), and leave a complete classification for future work.

4.3.1 Two By Two Matrices

As our general assumption is having just 0 or 1 on diagonal of the matrix, for the 2×2 matrices we must consider the possible cases for the only off-diagonal entry. If the off-diagonal is $*$ the problem is partition of a graph to cliques and independent sets, without any constraints; Theorem 3.3.13, describes all the finitely many minimal chordal obstructions.

If the off-diagonal is not $*$ then the matrix is without any asterisk entries. Theorem 3.3.5, implies there are only finitely many minimal chordal obstructions.

4.3.2 Three By Three Matrices with Zero Diagonal

In this section we study partitions of graphs into three independent sets. In other words we study 3×3 matrices with just 0's on diagonal. Here M has the general form of:

$$\begin{pmatrix} 0 & a & b \\ a & 0 & c \\ b & c & 0 \end{pmatrix}$$

If none of a, b, or c is asterisk, then M is a matrix without $*$ entries, and Theorem 3.3.5 implies there are only finitely many minimal chordal obstructions. Therefore we assume at least one off-diagonal entries is $*$. On the other hand if all of the off-diagonal entries are $*$ then Theorem 3.3.13 describes all the finitely many minimal chordal obstructions. Without loss of generality we suppose $b = *$ and $c \neq *$. If a is zero then we apply Lemma 4.2.2 and conclude by theorem, and transfer the problem to a two by two case. As discussed above, in section 4.3.1, there are only finitely many minimal obstructions. By symmetry, the same argument is valid when $c = 0$. It remains to consider, up to symmetry, the case when $a = *, c = 1$ (Theorem 4.3.1) and when $a = c = 1$ (Theorem 4.3.2).

Theorem 4.3.1. *Let* $M = \begin{pmatrix} 0 & * & * \\ * & 0 & 1 \\ * & 1 & 0 \end{pmatrix}$.

There are infinitely many minimal chordal obstructions for M-partition problem.

Proof. The structure of a graph which is M-partitioned is shown in Figure 4.1.

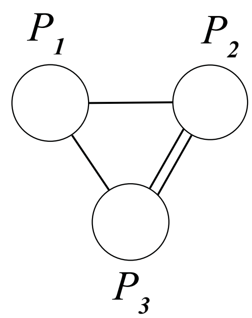


Figure 4.1: The structure of an M-partitioned graph according to Theorem 4.3.1.

Any graph $G(t)$, $t \geq 3$, from Figure 4.2 is a minimal chordal obstruction for this partition problem.

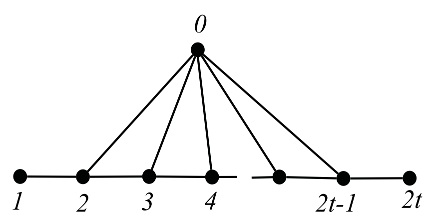


Figure 4.2: $G(t)$

First, we show that $G(t), t \geq 3$ is not M-partitionable. Suppose $G(t)$, $t \geq 3$ is M-partitionable. If 0 is in the first part, then $2, 3, \cdots 2t-1$, $(2t-1 \geq 5)$ must go to the second or third parts, since they are all adjacent to 0. Suppose 2 is in the second part then, adjacency of 2 and 3 implies 3 must be in the third part. The edge between 3 and 4 implies 4 to be in the second part; and the edge between 4 and 5, implies that 5 must be in the third part. But here vertices 2 and 5 are not adjacent. So it is not possible to have an M-partition for $G(t)$ while 0 is in the first part.

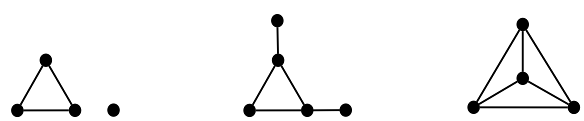
If 0 is in the second part, then its neighbours that can not map to the second part and due to their adjacencies must alternately go to the first and third part. Without loss of generality we can suppose that the even vertices are in the first part, and the odd vertices are in the third part. Now the first vertex can not map to the third part, since it is not adjacent to 0; and it can not not map to the first part since it is adjacent to 2. so no M-partition for $G(t)$ is possible with 0 in the second part. The same argument applies if 0 is in the third part.

It remains to show that $G(t)$'s are minimal obstructions. That if we omit any vertex of $G(t)$ then, it has an M-partition.

If 0 is deleted, the remaining graph is bipartite: we can map the even vertices to the first part, and the odd vertices to the third part. If 1 is deleted: we can map 0 to the second part, the odd vertices to the third part, and the even vertices to the first part. If $2t$ is deleted we can put 0 in the second part, the odd vertices in the first part, and the even vertices in the third part. If a vertex $1 < v < 2t$ is deleted, we can put 0 in the second part, then starting from 1, alternately put the remaining vertices in the first and third parts. $\square$

Theorem 4.3.2. *Let* $M = \begin{pmatrix} 0 & 1 & * \\ 1 & 0 & 1 \\ * & 1 & 0 \end{pmatrix}$.

A chordal graph G is M-partitionable if and only if it does not contain any of the graphs in Figure 4.3 as an induced subgraph.



(a) G_1 (b) G_2 (c) K_4

Figure 4.3: Minimal obstructions, Theorem 4.3.2 and Lemma 4.3.1

Proof. The structure of a graph which is M-partitioned is shown in Figure 4.4.

First we show that each of the graphs in Figure 4.3 is a minimal obstruction.

Since G_1 is not a bipartite graph and does not have a universal vertex, it is not M-partitionable according to the Lemma 4.3.1.

On the other hand deleting any vertex of the triangle would result in a bipartite graph,

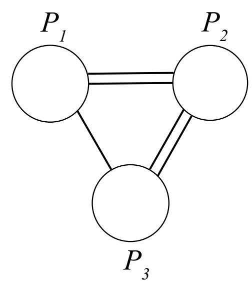


Figure 4.4: The structure of an M-partitioned graph according to Theorem 4.3.2.

that can be partitioned to the first and third parts. Also, deleting the isolated vertex, M-partition of the remaining graph (the triangle) can be obtained by putting a vertex in the first part, a vertex in the second part, and a vertex in the third part.

G_2 is not M-partitionable, because it is not bipartite and does not have a universal vertex (4.3.1).

G_2 is minimal, because if we delete any vertex of its triangle then the remaining graph is bipartite and partitionable to the first and third parts. If we delete any vertex of degree one, M-partition can be obtained by putting the vertex of the triangle which has degree three in the second part, the other vertices of the triangle to the first and third parts, separately; and the remaining vertex in the first part.

If K_4 is M-partitionable, then the pigeonhole principle implies that there must be a part with at least two vertices (there are four vertices and three parts); but as each part is an independent set two adjacent vertices of K_4 cannot be in the same part. On the other hand deleting any vertex of K_4, the remaining graph is a triangle. An M-partition can be depicted by putting each vertex in a different part.

Now we show that if the input graph G does not have any of the graphs in Figure 4.3 as an induced subgraph, it is M-partitionable. If G is a tree, it is a bipartite graph, too; so vertices can map to the first and third parts. Thus assume G contains a cycle, i.e., a triangle (G is chordal). Lemma 4.3.1 implies that G has a universal vertex v. We can map v to the second part. The remaining graph $G\backslash\{v\}$ is bipartite, therefore we can map its vertices to the first and third parts.

If $G\backslash\{v\}$ is not bipartite, then it must have a triangle. As v is a universal vertex in G, it is adjacent to all tree vertices of the triangle; therefore G must have an induced K_4, which in is contradiction wit assumption. $\square$

Lemma 4.3.1. *Assume that G is a non-bipartite chordal graph which does not contain G_1 or G_2 of Figure 4.3 as an induced subgraph. Then G has a universal vertex.*

Proof. A chordal graph G which is not bipartite has a triangle. We claim that at least one vertex of the triangle must be adjacent to all vertices of the graph. Each vertex of G is adjacent to at least one of the vertices of the triangle, otherwise we have an induced G_1 in G.

Any two vertices of G must have a common neighbour in the triangle. Indeed, suppose u and v do not have a common neighbour in the triangle. This u and v are not adjacent, or else u, v and their different neighbours in the triangle form a four cycle. If each of u and v has exactly one neighbour in the triangle we have an induced G_2 in G. If u has two neighbours a and b in triangle and v has one neighbour c then u, a, b which is a triangle is independent of v and these four vertices form a G_2. Therefore any two vertices of G has a common neighbour in the triangle.

Now we claim that one of the vertices of the triangle is universal. Suppose the triangle a, b, c does not have a universal vertex. Let p be the non-neighbour of a, and q be the non-neighbour of b, and r be the non-neighbour of c. Then p, q, r are independent of each other, and each of must have two neighbours in a, b, c triangle. Now the induced subgraph with vertices p, b, c, q, r forms a G_2. Therefore, one of the vertices of the triangle must be adjacent to every other vertex; and G has a universal vertex. $\square$

4.3.3 Three by Three Matrices with Ones on Diagonal

In this section we study partitions of graphs into three cliques. In other words we study 3×3 matrices with just 1's on diagonal. Here M has the general form of:

$$\begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix}$$

If none of a, b, or c is asterisk, then M is a matrix without $*$ entries, and Theorem 3.3.5 implies there are only finitely many minimal chordal obstructions. Therefore we assume at least one off-diagonal entry is $*$. On the other hand if all of the off-diagonal entries are $*$ then Theorem 3.3.13 describes all of the finitely many minimal chordal obstructions. Without loss of generality we suppose $b = *$ and $a \neq *$. If $c = 1$ we apply Lemma 4.2.3. As it is discussed in Section 4.3.1, there are only finitely many minimal obstructions. By symmetry, the same argument is valid when $a = 1$.

It remains to consider, up to symmetry, the case when $a = c = 0$ (Theorem 4.3.3), and when $a = *, c = 0$ (Theorem 4.3.4).

Theorem 4.3.3. *Let* $M = \begin{pmatrix} 1 & 0 & * \\ 0 & 1 & 0 \\ * & 0 & 1 \end{pmatrix}$.
There are only finitely many minimal chordal obstructions for the M-partition problem.

Proof. The structure of a graph which is M-partitionable is shown in Figure 4.5.

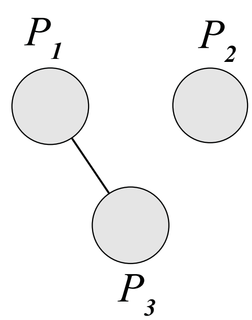


Figure 4.5: The structure of an M-partitioned graph according to Theorem 4.3.3.

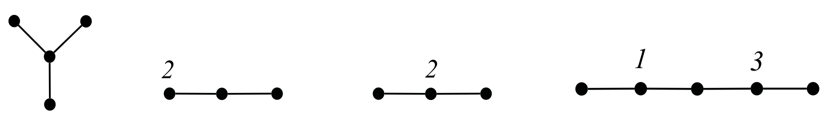
We shall prove the Theorem by considering chordal graphs with a given independence number α. Thus there are only finitely many minimal chordal obstructions, by classifying them according to their independence number.

If $\alpha \leq 2$, then input graph G can be covered by two cliques, since G is chordal and hence perfect graph. We can map the vertices of one clique to the first part, and the vertices of the other clique to the third part.

If $\alpha \geq 4$, then there must be at least four independent vertices in G. But $\bar{K}_4$ is a minimal obstruction, since there are three clique parts in this partition problem, and four independent vertices. Deleting one vertex of $\bar{K}_4$, partition of the remaining graph can be obtained by putting each vertex to a different part. It follows that $\bar{K}_4$ is the unique minimal chordal obstruction with independence number at least four.

The remaining case is when the independence number of G is three. So there are three independent vertices, say x, y and z. If G is M-partitionable then these vertices must go to different parts. Without loss of generality, we assume x goes to the first part and we pre-colour x by 1; and y goes to the second part and we pre-colour it by 2, and finally z goes to the third part and we pre-colour it by 3.

Now we show that if G with these pre-coloured vertices does not contain any of the minimal chordal pre-coloured obstructions in Figure 4.6 as an induced subgraph, then G is M-partitionable.



(a) O_1 (b) O_2 (c) O_3 (d) O_4

Figure 4.6: Chordal minimal pre-coloured obstructions with $\alpha = 3$.

The remaining vertices of G can be classified according to their adjacencies with x, y and z. Let S_{xyz} denote the set of vertices adjacent to x, y and z; let S_{xy} denote the set of vertices adjacent to x, y but not adjacent to z; and let S_x denote the set of neighbours of x that are adjacent to neither y, nor z. Similarly, we define S_{xz}, S_{yz}, S_y and S_z. As the independence number of G is three, there is no vertex in G independent of these three vertices, and as a result we can say $V(G) = \{xyz\} \cup S_x \cup S_y \cup S_z \cup S_{xy} \cup S_{xz} \cup S_{yz} \cup S_{xyz}$. Figure 4.7 shows this neighbourhood classification.

The chordality of G implies that each of S_{xyz}, S_{xy} S_{yz}, and S_{xz} is a clique. For example if S_{xy} is not a clique, then there are two non-adjacent vertices u, v in S_{xy}. This u, v and

x, y together form an induced four cycle, which is in contradiction to the chordality of G. Similar arguments can be made for S_{xyz}, S_{yz} and S_{xz}.

Furthermore, each of S_x, S_y, and S_z is also a clique. For instance if S_x were not a clique, there would be two non-adjacent vertices u and v in in S_x. Now u and v that are not adjacent neither to y nor to z form four independent vertices, which is contradiction. Similar arguments can be to show that S_y and S_z are cliques.

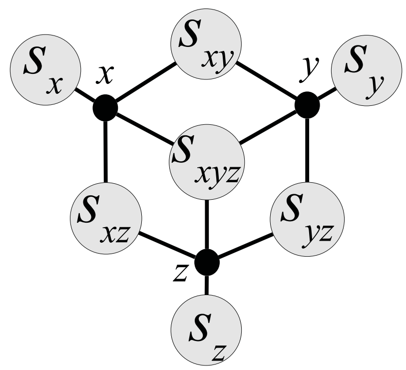


Figure 4.7: Adjacencies in G relative to the independent vertices x, y, z.

Note that this figure does not show the adjacencies between the classes S_x, S_{xy}, etc. Therefore the fact that there is no line between the circles does not mean they are independent.

Now note that the set $Sxyz$ must be empty, otherwise $v \in Sxyz$ and x, y, z form an induced O_2 from Figure 4.6. Also $S_{xy} = S_{yz} = \emptyset$; otherwise if there is a $v \in Sxy$ then yvx forms an induced O_2 from Figure 4.6.

Each vertex in S_{xz} must be adjacent to all vertices of S_x or all vertices S_y; otherwise there is a vertex $v \in S_{xz}$, a vertex $w \in S_x$, and a vertex $u \in S_y$ such that v is adjacent to neither w nor u; therefore w, x, v, z, u form a graph O_4.

Let A denote set of vertices of S_{xz} that are adjacent to all vertices in S_x, and B denote the remaining vertices of S_{xz}. Note that each vertex in B is adjacent to all vertices of S_z. An M-partition is obtained by putting the vertices of $\{x\} \cup S_x \cup A$ in the first part, vertices of $\{y\} \cup S_y$ in the second part, and vertices of $\{z\} \cup S_z \cup B$ in the third part. The set $S_x \cup A$ is a clique, by the definition of A; and they are all adjacent to x. So all of vertices that we map to the first part form a clique. Similarly $z \cup S_z \cup B$, vertices that we map to the third part, form a clique. Set S_y is independent of S_x and S_z, otherwise a vertex $v \in S_y$ and y

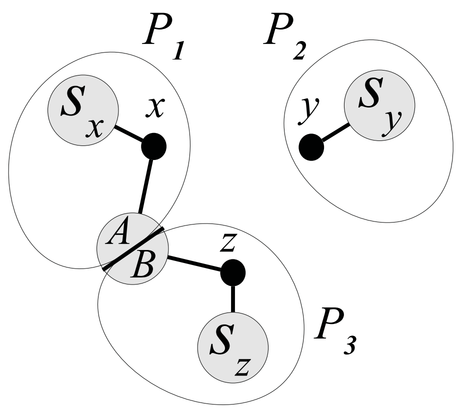


Figure 4.8: An M-partitioned chordal graph G according to Theorem 4.3.3.

and a vertex $w \in S_x$ (or $w \in S_z$) form an O_2; therefore the vertices that we map to the second part are independent from the vertices that we map the first and third parts. Recall that each of S_x, S_y, S_z is a clique.

Figure 4.8 shows an M-partition of G. □

Theorem 4.3.4. *Let* $M = \begin{pmatrix} 1 & 0 & * \\ 0 & 1 & * \\ * & * & 1 \end{pmatrix}$.
There are only finitely many minimal chordal obstructions for M-partition problem.

Proof. The structure of a graph which is M-partitionable is shown in Figure 4.9.

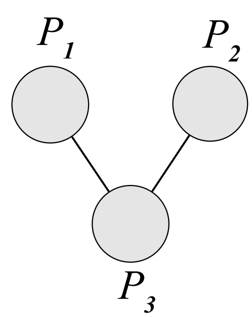
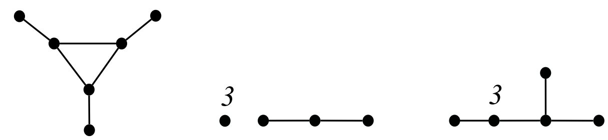


Figure 4.9: The structure of an M-partitioned graph according to Theorem 4.3.4.

We shall prove that there are only finitely many minimal chordal obstructions by classifying them according to their independence number. The proof is similar to the proof of the previous theorem, Theorem 4.3.3 when the independence number of the input graph G is at most two or at least four. So we consider G as a graph with independence number three.

As in the proof of Theorem 4.3.3, we pick three independent vertices x, y, z, and without loss of generality assume x goes to the first part, y goes to the second part, and z goes to the third part; thus we pre-colour x by 1, and y by 2, and z by 3.

Now we show that if G with these pre-coloured vertices does not contain any of the minimal obstructions in Figure 4.10 as an induced subgraph, then G is M-partitionable.



(a) H_1 (b) H_2 (c) H_3

Figure 4.10: Minimal obstructions when $\alpha = 3$ according to the Theorem 4.3.4.

First we show that each of the graphs in Figure 4.10 is a minimal obstruction.

Starting with H_1, if it is M-partitionable then then three independent vertices of degree one must go to different parts. This implies that each vertex of the triangle must also go to a different part (note that each vertex of the triangle is adjacent to just one non-triangle vertex and all parts are cliques in this partition problem). So there are two adjacent vertices such that one is in the first part and another one is in the second part, which is in contradiction. On the other hand H_1 is minimal. Suppose we eliminate a vertex of the triangle. The an M-partition of the remaining graph can be obtained by mapping the single vertex to the first part, two adjacent vertices to the second part and the other two adjacent vertices to the third part. This partition is shown in Figure 4.11(a). If we omit a non-triangle vertex, then the M-partition of the remaining graph, which is a bull graph, can be obtained by mapping a vertex of degree one to the first part, and mapping the other degree one vertex and its neighbour to the second part, and the remaining vertices to the third part. You can see this partition in Figure 4.11(b).

Then we show that H_2 is not M-partitionable. If H_2 is M-partitionable, three vertices of its path must go to the first and second parts, but this is in contradiction to the independence of the first and second parts. On the other hand H_2 is minimal. If we omit the labelled vertex, mapping two adjacent vertices to the first part and the remaining vertex to the third

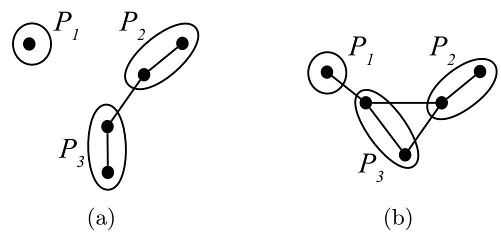


Figure 4.11: Minimality of H_1

part is an M-partition. Otherwise, if we omit a vertex of degree one, then an M-partition can be obtained by mapping two adjacent vertices to the first part; and if we omit the vertex of degree two, an M-partition can be obtained by mapping one of the single vertices to the first part, and the other one to the second part.

Finally we show that H_3 is not M-partitionable. If it is M-partitionable then three vertices of degree one must go to the different parts, such that the one which is adjacent to the pre-coloured vertex maps to the third part. But then the vertex of degree three must map to either the first part or the second part. It implies that a vertex in the first part is adjacent to a vertex in the second part, which is a contradiction. On the other hand H_3 is minimal. If we omit the pre-coloured vertex, then we have a partition for the rest of the graph by putting the single vertex in the first part, two adjacent vertices in the second part and the remaining vertex in the third part. If we omit a neighbour of the pre-coloured, an M-partition can be obtained by mapping the other neighbour of the pre-coloured vertex to the third part, and the other two vertices one in the first part and the other one in the second part. Also, if we omit a vertex of degree one which is not a neighbour of the pre-coloured vertex, then an M-partition can be obtained by mapping the neighbour of the pre-coloured vertex which is of degree two to the third part, and one of the remaining vertices to the first part and the other one to the second part.

So far we have shown that three graphs in the Figure 4.10 are minimal obstructions. Next we will show that any chordal graph G such that $\alpha(G) = \alpha = 3$ and does not contain any of these graphs, as an induced subgraph, is M-partitionable.

Assume x, y and z are three independent vertices in G. As in the proof of Theorem 4.3.3 we consider the classification of the remaining vertices of G according to their adjacencies with x, y, z, and define $S_x, S_y, S_z, S_{xy}, S_{xz}, S_{yz}, S_{xyz}$, as it is shown in Figure 4.7.

The following procedure presents an M-partition of G.

- We map x and the vertices in S_x to the first part.
- We map y and the vertices in S_y to the second part.
- We map z and the vertices in S_z and S_{xyz} to the third part.
- Let A consist of all vertices v in S_{xz} that are non-adjacent to some u in S_z.
  We map these vertices to the first part.
- Let B consist of all vertices v in S_{yz} that are non-adjacent to some u in S_z.
  We map these vertices to the second part.
- Let C consist of those remaining vertices of S_{xz} that are either adjacent to some vertex in $S_y \cup B$, or non-adjacent to some vertex in S_x.
  We map these vertices to the third part.
- Let D consist of those remaining vertices of S_{yz} that are either adjacent to some vertex in $S_x \cup A$, or non-adjacent to some vertex in S_y.
  We map these vertices to the third part.

⊳⊳ Let E be the remaining vertices in S_{xz}, i.e., the vertices in $S_{xz} \backslash A \cup C$.
Let F be the remaining vertices in S_{yz}, i.e., the vertices in $S_{yz} \backslash B \cup D$.

⊳⊳ We consider the vertices of E in three separate sets E_1, E_2, E_3; and also
the vertices of F in three separate sets F_1, F_2, F_3 and continue the procedure as follows:

- Let E_1 be the vertices in E which are non-adjacent to some vertex in D.
  We map these vertices to the first part.
- Let E_1 be the vertices in E which are non-adjacent to some vertex in D.
  We map these vertices to the first part.
- Let F_1 be the vertices in F which are non-adjacent to some vertex in C.
  We map these vertices to the second part.
- Let E_2 be the vertices in E which are non-adjacent to some vertex in F_1.
  We map these vertices to the third part.
- Let F_2 be the vertices in E which are non-adjacent to some vertex in E_2.
  We map these vertices to the second part.
- Let E_3 be the vertices in $E \backslash E_1 \cup E_2$. We map these vertices to the first part.
- Let F_3 be the vertices in $F \backslash F_1 \cup F_2$. We map these vertices to the third part.

To summarize this procedure, the first part consists of the vertices in $\{x\} \cup S_x \cup A \cup E_1 \cup E_3$;

the second part consists of the vertices in $\{y\} \cup S_y \cup B \cup F_1 \cup F_2$; and the third part consists of the vertices in $\{z\} \cup S_z \cup S_{xyz} \cup C \cup D \cup E_2 \cup F_3$. Now we need to check all the adjacencies and non-adjacencies in these parts according to M. Lemma 4.3.2 shows those vertices that map to the first part by this procedure, induce a clique. Corollary 4.3.1 shows those vertices that map to the second part by this procedure, induce a clique. Lemma 4.3.3 shows that those vertices that this procedure maps to the third part, also induce a clique. And finally Lemma 4.3.3 shows that the vertices that map to the first part are completely independent from those that map to the second part. □

Note that S_{xy} is empty. Otherwise suppose there is a vertex v in S_{xy}; then the vertices x, v, y, z induce an H_2. Therefore the modified structure of G according to the adjacencies to the three independent vertices is shown in Figure 4.12(a); but for the rest of the figures, to have a more clear illustration, we do not show the set S_{xyz}, also we show the sets S_{xz} and S_{yz} a little bigger as Figure 4.12(b).

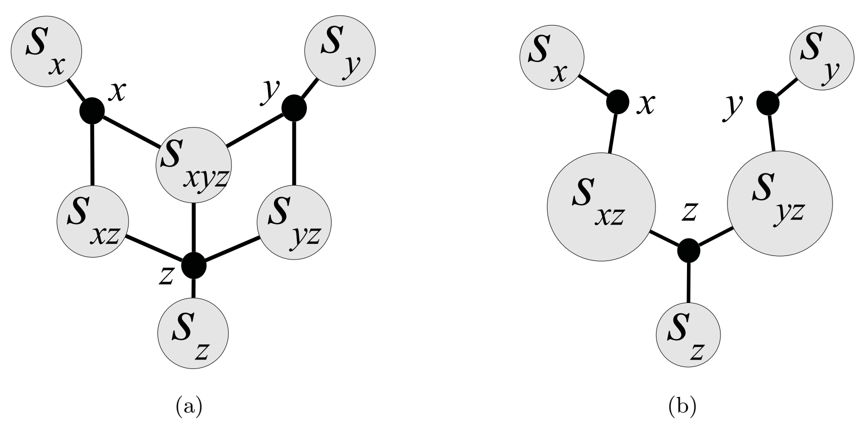


Figure 4.12

Lemma 4.3.2. *The vertices that the procedure maps to the first part induce a clique.*

Proof. We map the vertices of S_x, A and $E_1 \cup E_3 \subseteq E \subseteq S_{xz}$ to the first part. Chordality of G implies the vertices in S_x induce a clique, so do the vertices in S_{xz}: as it is argued in the proof of the previous theorem. So we must show that the vertices in $A \cup E_1 \cup E_3$ are completely adjacent to the vertices in S_x. Each vertex in E is adjacent to all vertices of

S_x, by the way we define C. Therefore $E_1 \cup E_3$ which is a subset of E, is also completely adjacent to S_x.

To show that A and S_x are completely adjacent, by contradiction suppose a vertex u in A is non-adjacent to a vertex v in S_x. Then u must have a non-adjacent vertex w in S_z. If v, w are adjacent then $vxuzw$ form a 5-cycle, which is not possible; and if v, w are not adjacent then four vertices v, u, w, y are independent, which is in contradiction to the assumption of $\alpha(G) = 3$. Therefore vertices mapped to the first part induce a clique. □

We can have a similar argument for the vertices that the procedure maps to the second part.

Corollary 4.3.1. *The vertices that the procedure maps to the second part induce a clique.*

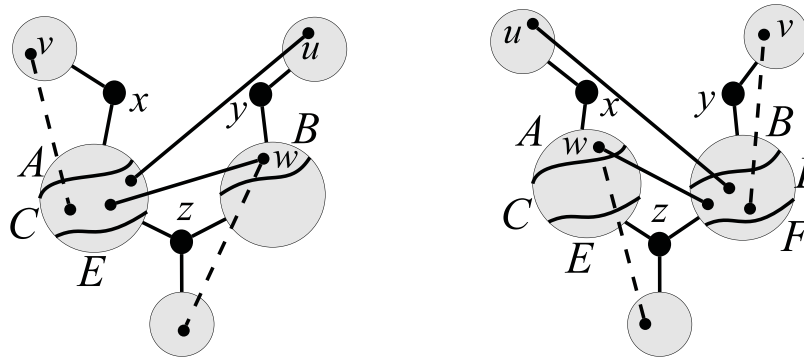
Lemma 4.3.3. *The vertices that the procedure maps to the third part induce a clique.*

Proof. As we have already shown, in the proof of Theorem 4.3.3 each of S_{xyz}, S_{xz}, S_{yz} and S_z is a clique. So it remains to show that all the edges between the sets S_{xyz}, S_z, C, D, E_2 and F_3 are present.

As G is chordal each vertex in S_{xyz} is adjacent to all vertices in S_{xz} and S_{yz} which implies that all edges between S_{xyz} and $C \cup E_2, D \cup F_3$ are present. Also by definition each vertex in $C \cup E_2$ or $D \cup F_3$ is adjacent to all vertices in S_z. So it suffices to show that all vertices in S_{xyz} and S_z are adjacent and so are the vertices in $C \cup E_2$ and $D \cup F_3$.

Suppose a vertex v in S_{xyz} is not adjacent to a vertex u in S_z. The induced subgraph on the vertices x, v, y, z, u induce an H_3, which is in contradiction to our assumption that G does not contain H_3. So all the vertices in S_{xyz} and S_z are adjacent.

Next we show that the vertices in C and D are all adjacent. But first we say a vertex in C is of *type one*, if it has a non-adjacent vertex in S_x; a vertex in C is of *type two*, if it is adjacent to some vertex in S_y; and a vertex in C is of *type three*, if it is adjacent to a vertex in B. Similarly we define type one, two and three for the vertices in D. In Figure 4.13(a) three types of vertices in C are shown; the vertex in C which is non-adjacent to v is of type one, the vertex which is adjacent to u is of type two, and the vertex which is adjacent to w is of type three. Similarly, In Figure 4.13(b) three types of vertices in D are show; the vertex in D which is non-adjacent to v is of type one, the vertex which is adjacent to u is of type two, and the vertex which is adjacent to w is of type three.



(a) (b)

Figure 4.13

Now we start showing all edges between C and D are present, by showing that each vertices of type one in C (D) is adjacent to all vertices of type one in D (C); each vertex of type one in C (D) is adjacent to all vertices of type two in D (C), and so on.

A vertex of type one in C is adjacent to all vertices of type one in D. Otherwise suppose v is a vertex of type one in C which is not adjacent to a vertex u of type one in D. As v is of type one, it has a non-adjacent vertex w in S_x, and u has a non-adjacent vertex t in S_y. You can see these vertices in Figure 4.14. Note w, t are non-adjacent, otherwise u, w, x and z induce an H_2. Also v and t are non-adjacent, otherwise t, x, v, z and u induce an H_3. Similarly we can argue that u and w are non-adjacent. Therefore u, v, w and t are pairwise non-adjacent, which implies $\alpha(G) = 4$. So we can say that all vertices of type one in C and D are adjacent.

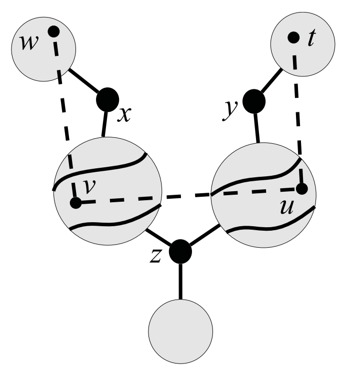


Figure 4.14

Next we show that it is not possible that C contains a vertex of type one when D contains a vertex of type two. Otherwise, let v be a vertex of type one in C, and u is a vertex of type two in D. The v has a non-adjacent vertex w in S_x and u has an adjacent vertex t in S_x. You can see these vertices in Figure 4.15. First note that w and t must be identical, otherwise if u, v are non-adjacent $wtuzvx$ is an induce 6-cycle, or if u, v are adjacent then $wtuvx$ is a 5-cycle. Now, if u, v are non-adjacent then $wuzvx$ is an induced 5-cycle; and if u, v are adjacent then $wuvx$ is an induced 4-cycle.

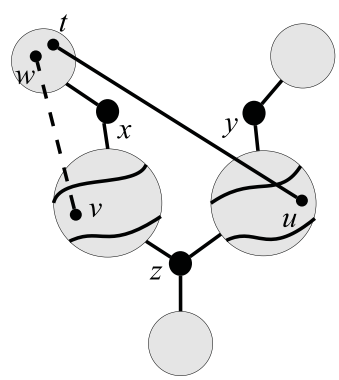


Figure 4.15

Now we show that type one vertices of C are adjacent to type three vertices of D. Suppose v is a vertex of type one in C and u is a vertex of type three in D. Let w be the vertex in A which is adjacent to u. By definition of vertices in A, there must a vertex t in S_z which is non-adjacent to w. You can see these vertices in Figure 4.16. If u and v are non-adjacent then, $wutv$ form an induced 4-cycle. So type one vertices of C and type three vertices of D are all adjacent.

Next we show that vertices of type two in C are adjacent to all vertices (type two and three) in D. Suppose v is vertex of type two in C, and u is a vertex in D. Note that as it is shown above u is not of type one, therefore u is adjacent to all vertices in S_y. As v is on type two then it has an adjacent vertex w in S_y. These vertices are shown in Figure 4.17. If u, v are not adjacent, then $vwuz$ induce a 4-cycle.

Here we show that vertices of type three in C are adjacent to all vertices of type three in D. Suppose v is a vertex of type three in C and u is a vertex of type three in D. Let v be adjacent to vertex w in B, and u be adjacent to a vertex t in A. You can see these vertices in Figure 4.18. If u, v are not adjacent then $vwut$ is a 4-cycle (in the following part

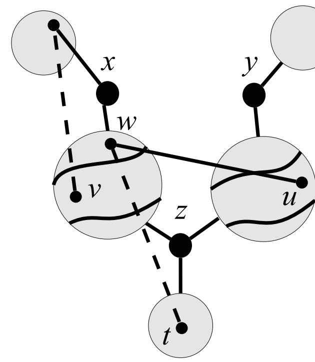


Figure 4.16

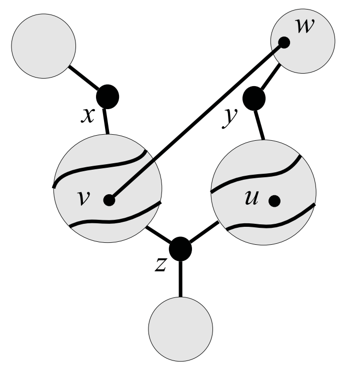


Figure 4.17

we show that the vertices in A and B are independent, so w, t are non-adjacent.

So in this way, we showed that the vertices in C and D are completely adjacent.

By definition each vertex in E_2 is adjacent to all vertices of D, because if a vertex v has a missing edge to D, the procedure put v in E_1. Similarly F_3 is adjacent to all vertices of C. Each vertex in F_3 is adjacent to all vertices in E_2, otherwise the procedure put the vertex of F with the missing edge to E_2, in F_2.

Therefore we proved the vertices that map to the third part have all the possible edges present. □

Lemma 4.3.4. *The vertices that the procedure maps to the first part are completely independent from the vertices that it maps to the second part.*

Proof. Vertices in S_x are independent from the vertices in S_y; otherwise $u \in S_x$ is adjacent to $v \in S_y$, but x, u, v and z induce an H_2. So S_x and S_y are independent. Also, A and S_y

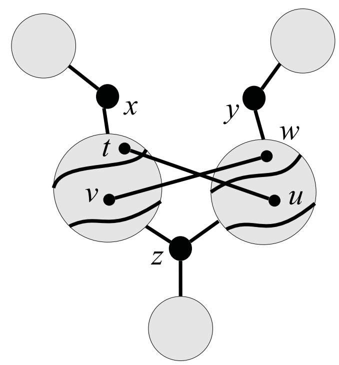


Figure 4.18

must be independent; otherwise $v \in A$ is adjacent to $u \in S_y$, and v, u, y and z induce an H_2.

Vertices in A are independent from the vertices in S_y; otherwise suppose adjacent vertices $v \in A$ and a vertex $u \in S_y$. Then x, v, u, y, z form an induced H_3.

Vertices in A are independent from the vertices in B; otherwise there are adjacent vertices $v \in A$ and $u \in B$. Suppose u and v have a common non-adjacent vertex w in S_z, then x, v, u, y, z and w induce an H_1. If u and v have different non-adjacent vertices in S_z, such that $w \in S_z$ is adjacent to u but non-adjacent to v, and $t \in S_z$ is adjacent to v but non-adjacent to u, then $vuwt$ induce a 4-cycle.

By definition vertices in E are independent form the vertices in S_y. So the subset $E_1 \cup E_3$ of E is also independent from S_y. Similarly $F_1 \cup F_2$ is independent from S_x.

Then we show that E_1 and F_1 are independent. By contradiction suppose a vertex $u \in E_1$ is adjacent to a vertex $v \in F_1$. The u must have a non-adjacent vertex $w \in D$, and v must have a non-adjacent vertex $t \in C$. And t and w are adjacent, as we showed in Lemma 4.3.3. But $uvwt$ induce a 4-cycle, which is not possible.

Next we show that E_1 and F_2 are independent. Again by contradiction we suppose a vertex $u \in E_1$ is adjacent to a vertex $v \in F_2$. By definition v must have a non-adjacent vertex $w \in E_2$, and w must have and adjacent vertex $t \in F_1$. In the previous part we show that E_1 and F_1 are independent, so w and t are not adjacent. Then $uvtw$ induce a 4-cycle.

The last thing is to show E_3 is independent from F_1 and F_2. The definition of E_2 implies that E_3 is independent from F_1. (If a vertex in $E \backslash E_1$ is adjacent to some vertex in F_1, our procedure put that vertex in E_2.) On the other hand, suppose $u \in E_3$ is adjacent to a vertex in F_2, say v. Then v must have a non-adjacent vertex $w \in E_2$, and w must have an adjacent vertex in F_1, say t. As we showed, by definition, E_3 and F_1 are independent, so u and t are non-adjacent. This implies that $uvtw$ induce a 4-cycle in G.

In this way we showed the vertices that our procedure maps to the first part, are independent from those that it maps to the second part. □

4.4 Conclusion

Classifying matrix partition problems according to their membership in the classes FOD, FMMO, $\mathcal{P}$ is an open problem for both general graphs and chordal graphs. Indeed the class of cographs is the largest graph classes these problems are solved. The partition problem for a cograph according to any matrix is in $\mathcal{P}$, and also in FMMO; by Theorem 3.2.1 it is also in FOD.

The classification of M-partition problems for general graphs, and small matrices M (of size 2×2, 3×3, 4×4), that whether the number of minimal obstructions is finite or infinite has been already studied [18]. However such classification has never been studied for chordal graphs. In our research we classified the M-partition problems for chordal graphs, according to the matrices of size 3×3 with constant diagonal. There were four cases left to consider. The first case is according to the matrix

$$M_1 = \begin{pmatrix} 0 & * & * \\ * & 0 & 1 \\ * & 1 & 0 \end{pmatrix}$$

The M_1-partition $\notin$ FMMO (Theorem 4.3.1), but it is polynomial time solvable.

The other three cases are as follows:

$$M_2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & * \\ 1 & * & 0 \end{pmatrix} \quad M_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & * \\ 0 & * & 1 \end{pmatrix} \quad M_4 = \begin{pmatrix} 1 & * & * \\ * & 1 & 0 \\ * & 0 & 1 \end{pmatrix}$$

We have proved that the M-partition problem for each of these three matrices is in FMMO (Theorems 4.3.2, 4.3.3, and 4.3.4). It is shown that the M-partition problem is polynomial for these cases.

Bibliography

[1] Claude Berge. *Farbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind, Wiss.* Z. Martin-Luther-Univ., Halle-Wittenberg Math.-Natur, Reihe 1961.

[2] Claude Berge. *The theory of graphs and its applications (Translated by Alison Doig).* Methuen & Co., John Wiley & Sons 1962.

[3] Kellog S. Booth and George S. Lueker. Testing of the consecutive ones property, interval graphs, and graph planarity using pqtree algorithms. pages 335–39. 1976.

[4] Andreas Brandstädt. Partitions of graphs into one or two independent sets and cliques. *Discrete Mathematics*, 152(1-3):47–54, 1996.

[5] Maria Chudnovsky, Gérard Cornuéjols, Xinming Liu, Paul D. Seymour, and Kristina Vuskovic. Recognizing berge graphs. *Combinatorica*, 25(2):143–186, 2005.

[6] Maria Chudnovsky, Neil Robertson, Paul D. Seymour, and Robin Thomas. Progress on perfect graphs. *Math. Program.*, 97(1-2):405–422, 2003.

[7] Vasek Chvátal. Star-cutsets and perfect graphs. *J. Comb. Theory, Ser. B*, 39(3):189–199, 1985.

[8] Derek G. Corneil, Stephan Olariu, and Lorna K. Stewart. The lbfs structure and recognition of interval graphs. *SIAM J. Discrete Math.*, 23(4):1905–1953, 2009.

[9] Derek G. Corneil, Yehoshua Perl, and Lorna K. Stewart. A linear recognition algorithm for cographs. *SIAM J. Comput.*, 14(4), 1985.

[10] Alain Cournier and Michel Habib. A new linear algorithm for modular decomposition. In *CAAP*, pages 68–84, 1994.

[11] Celina M. Herrera de Figueiredo, Sulamita Klein, Yoshiharu Kohayakawa, and Bruce A. Reed. Finding skew partitions efficiently. *J. Algorithms*, 37(2):505–521, 2000.

[12] Gabriel A. Dirac. On rigid circuit graphs. *Abh. Math. Sem. Univ Hamburg*, 25:71–76, 1961.

[13] Zdenek Dvorak, Daniel Kral, and Robin Thomas. Deciding first-order properties for sparse graphs. *FOCS 2010.*

[14] Jenö Egervéry. On combinatorial properties of matrices (hungarian with german summary). *Mat. Lapok*, 38:16–28, 1931.

[15] Tomás Feder and Pavol Hell. Matrix partitions of perfect graphs. *Discrete Mathematics*, 306(19-20):2450–2460, 2006.

[16] Tomás Feder and Pavol Hell. On realizations of point determining graphs, and obstructions to full homomorphisms. *Discrete Mathematics*, 308(9):1639–1652, 2008.

[17] Tomás Feder, Pavol Hell, and Winfried Hochstättler. Generalized colurings (matrix partitions) of cographs. *Trends in Mathematics*, pages 149–167, 2006.

[18] Tomás Feder, Pavol Hell, Sulamita Klein, and Rajeev Motwani. Complexity of graph partition problems. In *STOC*, pages 464–472, 1999.

[19] Tomás Feder, Pavol Hell, Sulamita Klein, and Rajeev Motwani. List partitions. *SIAM J. Discrete Math.*, 16(3):449–478, 2003.

[20] Tomás Feder, Pavol Hell, Sulamita Klein, Loana Tito Nogueira, and Fábio Protti. List matrix partitions of chordal graphs. *Theor. Comput. Sci.*, 349:52–66, 2005.

[21] Tomás Feder, Pavol Hell, and Wing Xie. Matrix partitions with finitely many obstructions. *Electr. J. Comb.*, 14(1), 2007.

[22] Tomás Feder and Moshe Y. Vardi. Homomorphism closed vs. existential positive. In *LICS*, pages 311–320, 2003.

[23] D.R. Fulkerson and O.A. Gross. Incidence matrices and interval graphs. In *Pacific J. Math.*, pages 835–855. Fund. Math., 1965.

[24] Fanica Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory*, 16:47–76, 1974.

[25] Alain Ghouila-Houri. Caracterisation des graphes non orientes dont on peut orienter les arretes de maniere a obtenir le graphe d'une relation d'ordre. In *C. R. Acad. Sci.*, pages 1370–1371, 1962.

[26] Gilmore and A.J. Hoffman. A characterization of comparability graphs and interval graphs. In *Pacific J. Math.*, pages 539–548. Canadian J. Math. 16, 1964.

[27] Martin C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Elsevier Science, 2nd edition edition, 2004.

[28] Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications (Texts in Theoretical Computer Science. An EATCS Series).* Springer, 2007.

[29] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, 1988.

[30] Michel Habib, Ross M. McConnell, Christophe Paul, and Laurent Viennot. Lex-bfs and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theor. Comput. Sci.*, 234(1-2):59–84, 2000.

[31] Michel Habib and Christophe Paul. A simple linear time algorithm for cograph recognition. *Discrete Applied Mathematics*, 145(2):183–197, 2005.

[32] Peter L. Hammer and Stéphane Földes. Split graphs. In *Eighth Southeastern Conference on Combinatorics, Graph Theory and Computing*, pages 311–315, 1977.

[33] Peter L. Hammer and Bruno Simeone. The splittance of a graph. *Combinatorica*, 1(3):275–284, 1981.

[34] Pavol Hell, Sulamita Klein, Loana Tito Nogueira, and Fábio Protti. Partitioning chordal graphs into independent sets and cliques. *Discrete Applied Mathematics*, 141(1-3):185–194, 2004.

[35] Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms.* Oxford University Press 2004.

[36] Pavol Hell and Jaroslav Nešetřil. On the complexity of h-coloring. *J. Comb. Theory, Ser. B*, 48:92–110, 1990.

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

[38] Sulamita Klein and Celina M. H. De Figueiredo. The np-completeness of multi-partite cutset testing. *Congr. Numer*, 119:217–222, 1996.

[39] Dénes König. Gráfok és alkalmazásuk a determinánsok és a halmazok elméletére. *Matematikai és Természettudományi Értestõ*, (34):104–119, 1916.

[40] Dénes König. Graphen und matrizen. *Mat. Lapok*, 38:116–119, 1931.

[41] Norbert Korte and Rolf H. Möhring. An incremental linear-time algorithm for recognizing interval graphs. *SIAM J. Comput.*, 18:68–81, 1989.

[42] C. G. Lekkerkerker and J. C. Boland. Representation of a finite graph by a set of intervals on the real line. Discourse Processes: Advances in Research and Theory, pages 45–64. Fund. Math., 1962.

[43] László Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Mathematics*, 2(2):253–267, 1972.

[44] Ross M. McConnell and Jeremy Spinrad. Linear-time modular decomposition and efficient transitive orientation of comparability graphs. In *SODA*, pages 536–545, 1994.

[45] Ross M. McConnell and Jeremy Spinrad. Linear-time transitive orientation. In *SODA*, pages 19–25, 1997.

[46] Donald J. Rose and Robert E. Tarjan. Algorithmic aspects of vertex elimination. In *STOC*, pages 245–254, 1975.

[47] Donald J. Rose, Robert E. Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976.

[48] Benjamin Rossman. Existential positive types and preservation under homomorphism. In *LICS*, 2005.

[49] Robert E. Tarjan. Algorithm aspects of vertex elimination on directed graphs. In *Maximum cardinality search and chordal graphs*. Stanford University, unpublished Lecture Notes.

[50] Robert E. Tarjan. Decomposition by clique separators. *Discrete Mathematics*, 55(2):221–232, 1985.

[51] Alan C. Tucker. Coloring graphs with stable sets. *Combin Theory*, 34:258–267, 1983.

[52] Narayan Vikas. Computational complexity of graph compaction. In *Tenth Annual ACM SIAM Symposium on Discrete Algorithms*. Baltimore, MD, 1999.

[53] Douglas B. West. *Introduction to graph theory.* Prentice Hall, 2001.

[54] Sue Whitesides. An algorithm for finding clique cut-sets. *Inform. Process. Lett.*, 12:31–32, 1981.

[55] Sue Whitesides. A method for solving certain graph recognition and optimization problems, with applications. *Ann. Discrete Math*, 21:281–297, 1984.