University of Alberta

Library Release Form

Name of Author: James Nastos

Title of Thesis: $(P_5, \overline{P}_5)$-free Graphs

Degree: Master of Science

Year this Degree Granted: 2006

James Nastos
1128 113 Street S.W.
Edmonton, Alberta
Canada, T6W 1P5

Date: ____________________

University of Alberta

$(P_5, \overline{P}_5)$-free Graphs

by

James Nastos

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Spring 2006

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled $(P_5, \overline{P}_5)$**-free Graphs** submitted by James Nastos in partial fulfillment of the requirements for the degree of **Master of Science**.

Ryan Hayward
Supervisor

Lorna Stewart

Mazi Shirvani
External Examiner

Date: ____________________

Abstract

Cographs are graphs arising from various applications and the study of these has been pivotal in the development of various algorithmic graph theory techniques and properties, such as modular decomposition.

Perfectly orderable graphs do not typically arise from application modeling. Instead, they are defined as those graphs with a desirable property making certain optimization problems solvable on them in a simple and efficient manner.

(P_5, $\overline{P}_5$)-free graphs generalize cographs and have attracted much attention in recent years. The class of (P_5, $\overline{P}_5$, C_5)-free graphs is a self-complementary class of perfectly orderable graphs on which several optimization problems are solvable in linear time, yet the recognition problem for this class has no known algorithm faster than $\Theta(n^3)$-time. When applying modular decomposition, recognizing (P_5, $\overline{P}_5$, C_5)-free graphs is sufficient to recognize (P_5, $\overline{P}_5$)-free graphs.

We investigate the structure of (P_5, $\overline{P}_5$)-free and (P_5, $\overline{P}_5$, C_5)-free graphs. This thesis reveals some properties, gives counterexamples, and develops some conjectures concerning this structure.

Acknowledgements

Engaging in graduate studies at the University of Alberta has been both an honour and a privilege, so I wish to thank everyone contributing to this opportunity. I am indebted to my family, former teachers and professors and all my supportive friends.

The faculty and graduate students in the department of Computing Science have made my (too many) years here a pleasurable and rewarding experience. The friendliness and openness of the researchers with whom I associated made scientific research a joy beyond my expectations. I would like to thank my fellow graduate and undergraduate students who have always been available for general chatting, encouragement, discussion of research, helping with technical issues and sharing of thesis woes. I wish to acknowledge W. Sean Kennedy's graph-drawing program, which I found very useful in my thesis work.

I thank my supervisor, Ryan B. Hayward, for his patience with me, confidence in my abilities and for taking me as his student. Lunch outings were always a delight, especially when he shared personal stories of his days as a graduate student. The financial assistance during the terms he supported me was generous and of great help. Working alongside him has been an enlightening and humbling experience; an hour of his thoughts, observations and conjecture-forming would often surpass a week's worth of my work. Our trip to the Perfect Graph Theorem workshop in Palo Alto will always be remembered. Above all, I appreciate the eternally-frank and often-jocular relationship we have had.

Finally, I thank Vivian for moving with me to Edmonton, whose companionship I would have longed for had she not.

Contents

List of Figures

Chapter 1

Introduction to Graphs and Graph Classes

1.1 Graphs

A graph can be thought of as a pictorial representation of objects and relations between the objects, typically by denoting objects with dots (called *vertices*) and relations between objects with lines (called *edges*) joining the dots.

To formally define a graph, the vertices are denoted by a set $V(G)$ and the edges are denoted by the set $E(G)$ which is a set of pairs of elements from $V(G)$. When the graph in context is clear, $V(G)$ and $E(G)$ are often simplified to V and E. To specify a graph with vertex set V and edge set E, we write $G = (V, E)$. Graphs can model a wide variety of problems and scenarios. For instance, the vertices could be communication centres and the edges could represent communication channels, such as a network of computers or a collection of cities connected by train tracks. Visual examples of graphs are available in Figure 2.3, and examples of objects and relations that are represented by graphs are discussed in Section 1.3.1.

The terms we use are standard, and will be defined later in Section 1.1.1. Chapter 1 covers some fundamentals in graph theory algorithms and analysis, and also provides examples which motivate the general study of graph classes. Chapter 2 reviews some important notions in algorithmic graph theory that are necessary to fully appreciate and understand the approach taken in the study of (P_5, $\overline{P}_5$)-free graphs following. Chapter 3 provides motivation for the study of (P_5, $\overline{P}_5$)-free graphs by showing that they are a relevant and important graph class to study, particularly for the associated recognition problem. Chapter 4 reviews three papers which characterize and recognize classes related to (P_5, $\overline{P}_5$)-free graphs: (P_5, $\overline{P}_5$, bull)-free graphs, the semi-P_4-sparse graphs, and (P_5, $\overline{P}_5$)-sparse graphs. Chapter 6 presents new work on prime (P_5, $\overline{P}_5$)-free and (P_5, $\overline{P}_5$, C_5)-free graphs by categorizing the vertices depending on how they exist in induced P_4s in the graph. Chapter 7 recaps and concludes our thesis.

We will quickly describe the goal of the thesis here, noting that the following will be discussed in further detail in the chapters to come. The task of recognizing a graph class involves deciding whether a graph belongs to the class in question. Modular decomposition, discussed in Chapter 2, allows us the liberty of restricting our effort to recognizing those graphs in the (P_5, $\overline{P}_5$)-free and (P_5, $\overline{P}_5$, C_5)-free classes without modules. Split graphs (discussed in Section 3.1.1) are an easily-recognizable subclass of these two classes, and so we can remove these graphs from our consideration. A theorem of Hayward, Hougardy and Reed [31] tells us that every vertex in a non-split prime graph is in an induced P_4, and so, in an attempt to characterize the structure of prime (P_5, $\overline{P}_5$)-free and (P_5, $\overline{P}_5$, C_5)-free graphs, we partition the vertex set into vertices that exist as the ends of P_4s, the middle of P_4s, or both. Some simple properties follow, such as the end-only vertices are the simplicial vertices in the graph and so necessarily form an independent set. Similarly, the mid-only vertices are the co-simplicial vertices and so always form a clique. The remaining vertices are classified into several different types depending on how they relate to the end-only and mid-only vertices, and properties of each type are investigated. Even though many of these properties seem to be local to a vertex and its neighbourhood, they lead to some theorems and conjectures on the global structure of prime (P_5, $\overline{P}_5$)-free and (P_5, $\overline{P}_5$, C_5)-free graphs.

1.1.1 Notation and Definitions

In this subsection we present many of the basic graph theoretic definitions we will use in this thesis. A reader familiar with the field of algorithmic graph theory will be able to skip this section without any loss of continuity.

For vertices u and v in the vertex set V of a graph G, we say that u and v are *adjacent* is $\{u, v\}$ is in the edge set $E(G)$. We also say that vertex u *sees* vertex v if u is adjacent to v, and u *misses* v otherwise. We say an edge $\{u, v\}$ is *adjacent* to vertex u and to vertex v. We write $G = (V(G), E(G))$ to describe the (undirected) graph G with vertex set $V(G)$ and edge set $E(G)$. If the elements of E are ordered pairs (u, v) instead of sets $\{u, v\}$, then G is a *directed graph*. Unless otherwise stated, any graph mentioned in this thesis will be an undirected graph. When the context is clear, V and E will be used in place of $V(G)$ and $E(G)$. The *size* of a graph is the cardinality of V.

The *open neighbourhood of a vertex* v, denoted $N(v)$, is the set of all vertices adjacent to v. The *closed neighbourhood of a vertex* v, denoted $N[v]$, is $N(v) \cup \{v\}$. The *nonneighbourhood of* v is the set $V - N[v]$. We call elements of $N(v)$ (respectively, $V - N[v]$) the *neighbours* (respectively, *nonneighbours*) of v.

A *subgraph* H of a graph G is a graph (V_H, E_H) where V_H is a subset of V and E_H is a subset of E. Given a subset S of vertices of a graph $G = (V, E)$, the *induced subgraph on* S

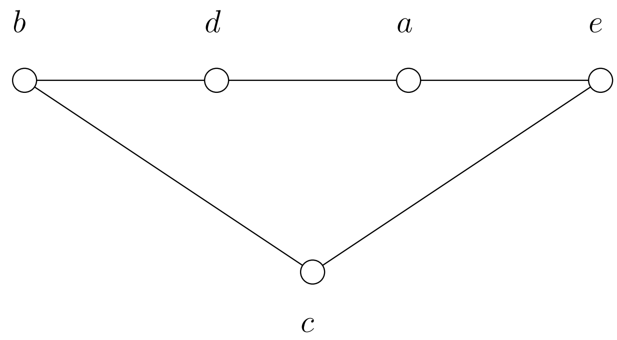

Figure 1.1: A cycle

is the graph (S, E_S) where an edge $\{u, v\}$ is in E_S if and only if u and v are in S and $\{u, v\}$ is in E. We will use $H \subseteq G$ to mean H is an induced subgraph of G.

Two graphs G_1, G_2 are *isomorphic* if there exists a bijection $f : V(G_1) \mapsto V(G_2)$ such that $\{u, v\}$ is in $E(G_1)$ if and only if $\{f(u), f(v)\}$ is in $E(G_2)$. The *complement* $\overline{G}$ of a graph G is the graph $(V, \overline{E})$ where for every pair of distinct vertices $u, v \in V$ we have $\{u, v\} \in \overline{E} \Leftrightarrow \{u, v\} \notin E$.

A *path* is a sequence of distinct vertices $v_1, v_2, \ldots, v_k$ such that $\{v_i, v_{i+1}\} \in E(G)$ for $i = 1 \ldots k-1$. Note that this definition uses an ordered sequence of the vertices in a path. For instance, in Figure 1.1 b, c, e, a is a path while b, c, d is not. For a path $v_1, v_2, \ldots, v_k$, an edge $\{v_i, v_j\}$ for $|i - j| \neq 1$ is called a *chord*. A path is a *chordless path* (equivalently, an *induced path*) if it has no chord. An induced path on k vertices is denoted P_k. A *cycle* is a sequence of vertices $v_1, v_2, \ldots, v_k$ such that $\{v_i, v_{i+1}\}$ is an edge for $i = 1 \ldots k-1$ as well as $\{v_k, v_1\}$. In Figure 1.1 a, d, b, c, e is a cycle while a, b, c, d, e is not. For a cycle $v_1, v_2, \ldots, v_k$, any edge $\{v_i, v_j\}$ for $i - j \not\equiv \pm 1$(modulo k) is called a *chord*. A cycle is a *chordless cycle* (equivalently, an *induced cycle*) if it has no chord. A *hole* is a chordless cycle on five or more vertices. Chordless cycles on k vertices are denoted C_k, and the complement of a hole is referred to as an *antihole*.

Given a path $v_1, \ldots, v_k$ we call the vertices v_1 and v_k the *endpoints* of the path. Any vertex in the path which is not an endpoint is called a *midpoint* of the path. A graph G is *connected* if for every pair of vertices v and w there exists a path in G with endpoints v and w. A graph is *disconnected* if it is not connected. The *(connected) components* of G are the subgraphs of G which are maximally connected ("maximal" here meaning with respect to subgraph inclusion.) The *wings* of a $P_{k \geq 3}$ are the two edges adjacent to the endpoints.

A subset H of the vertices V of a graph is called a *stable set* (also known as an *independent set*) if every pair of vertices in H are nonadjacent, and a *clique* (also known as a *complete graph*) if every pair of vertices of H are adjacent. A (*vertex*) *colouring* of a graph is a function $f : V \rightarrow \{1, 2, ..., k\}$ such that $\{u, v\} \in E \Rightarrow f(u) \neq f(v)$. The *size* of the colouring is the value k. The *clique number* (or *clique size*) of a graph G, denoted $\omega(G)$, is the size of the largest induced subgraph of G that is a clique. Similarly, the *stability number* (or *independence number*) of a graph G, denoted $\alpha(G)$, is the size of the largest induced subgraph of G that is a stable set. For simplicity, we will say "a graph *is* a certain other graph" in place of "a graph is isomorphic to a certain other graph." The *chromatic number* of a graph G, denoted $\chi(G)$, is the size of the smallest colouring of G.

For more details of the above concepts and for other graph theoretic definitions not mentioned here, refer to [23] or [60].

1.2 Optimization and Complexity

The reader is assumed to have prior exposure to basic complexity theory including algorithm time/space analysis, polynomial time algorithms and the theory of NP-Completeness, which are concepts covered in a standard introductory undergraduate algorithms course. A suitable introduction to these topics is covered in [10]. A deeper, more advanced treatment can be found in [49].

The problems of finding a maximum independent set, maximum clique, and minimum colouring of a graph are all NP-hard and so polynomial-time algorithms to solve these are currently unknown and may not even exist. As a result, one is often interested in solving these problems on a certain class of graphs, perhaps general enough for certain applications but restricted enough so that these problems can be efficiently solved. One such class of graphs is the set of *perfect graphs*, defined as those graphs G such that for every induced subgraph $H \subseteq G$, we have $\chi(H) = \omega(H)$. In 1977, Grötschel, Lovász and Schrijver [25], found that the above-mentioned optimization problems can be solved in polynomial time on perfect graphs. These algorithms are not considered efficient for use in applications. They rely on Khachiyan's ellipsoid method [42] from linear programming, an algorithm that is rarely preferred over the simplex method and its variants despite the polynomial runtime of the ellipsoid method [8]. Finding efficient optimization algorithms in both theory and practice is one reason perfect graphs have been extensively studied, as well as many subclasses of perfect graphs.

As is common in graph theory, for a graph $G = (V, E)$, n refers to $|V(G)|$ and m refers to $|E(G)|$. Labeled graphs (graphs whose vertices are labeled from 1 to n) are commonly represented either as an *adjacency matrix* or as an *adjacency list*.

The adjacency matrix M of a graph G is a matrix of dimension $n \times n$ whose rows and columns are indexed by the vertices of the graph and entry $M[i, j] = 1$ if vertices with labels i and j are adjacent in G, and $M[i, j] = 0$ otherwise. M is thus a symmetric {0,1}-matrix with main diagonal all zeroes, and so even with the removal of redundant information, this representation requires $\Theta(n^2)$ space. Testing the adjacency of two vertices takes constant time. In general, all algorithm runtimes mentioned are with respect to the RAM model of computation using a single processor machine. Algorithm runtimes will always be given as a measure of the worst-case analysis, unless otherwise mentioned.

The adjacency list of a graph is a list of rows indexed by the vertex labels. The i^{th} row is a list of all vertices adjacent to vertex i. This representation of a graph requires $\Theta(m+n)$ space. Testing the adjacency of two vertices by scanning these lists $O(n)$ time.

Algorithms are, unless otherwise noted, described in a manner which is representation-independent and the analysis of algorithms will assume constant-time adjacency testing. This is a reasonable assumption, as creating a $O(n^2)$-space representation of a graph with constant-time adjacency testing from an adjacency list is a standard textbook problem [1].

1.3 Graph Classes

The study of graph classes and algorithmic graph theory has flourished over the past fifty years, resulting in various books surveying the general field of graph classes (for example, the survey by Brandstädt, Le and Spinrad [3], or the book by Golumbic [23]) as well as books on specific graph classes (for example, tolerance graphs [24] or perfect graphs [50].) In this section we show how some graph classes are introduced, both from applications and from the natural generalization or restriction of other classes.

1.3.1 Classes from Application Modeling

Consider an individual trying to decide on what lectures to attend during a one-day conference. Every lecture at this conference spans a contiguous time interval and can be of any length. Two lectures are said to *conflict* if their time frames overlap, and so this individual can only attend non-conflicting lectures. We are interested in finding a largest set of non-conflicting lectures. Thus, if we create a graph whose vertex set is the set of lectures on that day and create an edge between two vertices if and only if the corresponding lectures conflict, then the task is solved by finding a maximum independent set in the resulting graph. If, instead, the individual wants to hire note-takers, and the fewest number of such note-takers, so that every lecture can be recorded, then we are interested in the size of the maximum clique of the graph as this will be the largest number of lectures that are occurring at any one instant in time. An optimal colouring of the graph will provide the schedule each note-taker should take, as each colour class will provide a conflict-free list of lectures.

As noted above, maximum independent set and clique are NP-hard optimization problems. However, since the graph created for this application was constructed from a specific structure (time intervals on a single time line,) the resulting graph will have some exploitable properties. For instance, if there exists any cycle of size four or more in this graph, then that cycle must have a chord. We call graphs obtained (as described in the preceding paragraph) from the intersection of intervals on a line *interval graphs*. The NP-hard problems of maximum clique, maximum independent set, minimum colouring all become polynomial-time solvable when the graph considered is an interval graph [26].

Now consider this individual allowing a few minutes to be spared at the beginning or end of each lecture, since sometimes arriving several minutes late does not affect the overall value of attending the lecture. In our problem formulation, we associate a *tolerance* with each time interval. If we define two intervals to be in conflict if and only if their intersection size is greater than at least one of the two tolerances, then the resulting graphs obtained from this problem are called *tolerance graphs*. See [24] for more information on these graphs.

1.3.2 Classes from Generalizations or Specifications

Here we describe a class of graphs and derive from it other graph classes whose definitions may have seemed artificial when removed from the context of the original graph class.

To solve a problem such as MAXIMUM CLIQUE on a graph G, it is clear that if the graph is disconnected, with components C_1 and C_2, then we may treat each connected component separately, find $\omega(C_1)$ and $\omega(C_2)$, and use the maximum of each as $\omega(G)$. This provides a simple decomposition into smaller problems. We can take this further by noting that if we are looking for a clique in a graph, then this is equivalent to finding an independent set in the complement of the graph. We further decompose our problems by now taking complements of each component and decomposing these into their connected components while keeping in mind that any maximum independent set of a graph is formed by taking the union of the maximum independent sets of its components.

Such a decomposition is natural to consider and is useful for a simple approach to solving some potentially hard problems. A problem, of course, occurs when the complement of a large graph is also connected and so no further decomposition is available. We henceforth define a class of graphs for which this never happens: G is a *complement reducible graph* (or simply a *cograph*) if it can be decomposed in the above way until every subproblem is reduced to a single vertex.

The class of cographs is a widely-studied class that arise in applications. A simple characterization of such graphs is through a *forbidden induced subgraph characterization*, namely, a list of graphs that never appear as induced subgraphs. For instance, the interval graphs never contain a chordless four-cycle, and so C_4 is one forbidden induced subgraph of interval graphs. For interval graphs in particular, there exists an infinite list of minimal such forbidden induced subgraphs. On the other hand, this list is very short for cographs: G is a cograph if and only if G is P_4-free [55].

Since restricting P_4s results in useful structural properties in the graphs, many cograph generalizations have been considered. For instance, Hoàng [33] defined P_4-*sparse* graphs as those graphs with the property that every set of five vertices induces at most one P_4.

Jamison and Olariu [39] define P_4-*reducible* graphs as those graphs such that every vertex is contained in at most one P_4. Many other similar graph classes have been defined to restrict the P_4s in some way, such as P_4-*extendible* graphs [40], P_4-*lite* graphs [41], and P_4-*laden* graphs [22]. Later, we will discuss *semi* P_4-*sparse* graphs, introduced in [19]. All these classes lead to useful decomposition schemes and generalize cographs (in the sense that if a graph is a cograph then it also belongs to any of these classes.)

1.4 Recognition

Now that we are aware of many graph classes, a natural question should be asked: Given a graph, is it a member of a particular graph class? This is called the *recognition problem* for a graph class. For the above cases, cographs (and all the other P_4-restricting classes mentioned) can be recognized in linear time. It can be the case that a recognition problem is NP-complete, as with the case of perfectly orderable graphs [47]. Interval graphs can be recognized in linear time, while tolerance graph recognition is still unknown to be polytime solvable.

The purpose of this thesis is to explore the recognition of $(P_5, \overline{P}_5)$-free graphs, which are introduced and discussed in Chapter 3. To the best of the author's knowledge, this is performed in $O(n^3)$ time using the HHD-free graph recognition algorithm [36], or $O(nm + m^2)$ time using the Meyniel graph recognition algorithm [52]. A deeper structural understanding often helps improve an algorithm, and structural results on $(P_5, \overline{P}_5)$-free graphs are presented in Chapter 6.

Chapter 2

Graph Tools

2.1 Lexicographic Breadth-First Search

The standard breadth-first (BFS) search algorithm is well known and often taught in any introductory algorithms course, so is omitted here. Refer to [10] for a description of BFS. Here, we describe a variant of BFS called *Lexicographic Breadth-First Search* [53] (LexBFS). LexBFS is an algorithm producing an ordering of the vertices $v_1, v_2, \ldots, v_n$ of a graph which is a BFS ordering but of a specific type. The algorithm begins as a normal BFS, starting at an arbitrary start vertex which will become vertex v_1, and all vertices are intialized with an empty label. For every neighbour u of of v_1, concatenate the label '1' to the end of the label of u. Vertex v_1 is marked as having been *seen* or *visited*, and of the unvisited nodes, we select a vertex of lexicographically strongest label (at this point, a vertex with label '1' is the "strongest" label.) Visiting one of these vertices and calling it v_2, we concatenate the label '2' to (the end of) all of the labels of v_2's neighbours. The earliest labels are considered lexicographically strongest (i.e. '1' is stronger than '2') and any non-empty label is considered stronger than an empty label. The possible labels of a vertex in our graph at this point are '12', '1', '2', '', and these are presented in a lexicographically decreasing order. LexBFS chooses the next vertex to visit as the vertex with the strongest lexicographical label. This process is continued until all nodes are visited. The labelling scheme essentially serves as a tie-breaking measure with respect to vertex choices in the standard BFS. In the case that two or more vertices share the lexicographically strongest label, any vertex with the strongest label may be chosen arbitrarily.

As one more example of a lexical comparison, label '125' comes before label '13456', since '125' is stronger in its second component. A search similar to BFS, called *maximum cardinality search* (MCS), will favour the longer labels over the shorter labels regardless of what the contents of the labels are. In the case that two or more vertices have the longest labels, an arbitrary vertex of longest label may be chosen next. The term "maximum cardinality" refers to the choice of picking a vertex whose set of already-visited neighbours is of maximum cardinality. For some classes of graphs, it has been shown that an MCS is equivalent to a LexBFS ordering [53].

MCS and LexBFS can be implemented to run in linear ($\mathrm{O}(m+n)$) time. See [53] for details.

2.2 Greedy Colouring

We defined earlier the concept of graph colouring and mentioned that computing an optimal (minimum) colouring is NP-hard. Here, we present a simple algorithm to colour a graph in a not-necessarily-optimal manner.

We will equate colours with the integers $1, 2, \ldots, k$, and use the term 'smallest' colour to mean the colour with smallest corresponding integer. One obvious colouring method is to colour vertex v_i with colour i. This provides a trivial upper bound to the chromatic number ($\chi(G) \leq |V|$,) and this upper bound is realized when the input graph is a clique. Note that any valid colouring of the graph provides an upper bound on the chromatic number. The *greedy colouring algorithm*, in its most general form, takes an arbitrary ordering of the vertices of a graph and proceeds to visit the vertices in this order, colouring each vertex with the lowest colour available (that is, the lowest colour not already assigned to a neighbour of that vertex.) This process runs in $\mathrm{O}(m+n)$ time and provides a valid colouring size less than or equal to $|V|$. Graphs exist with specific vertex orders for which the difference between the greedily obtained colouring size and the chromatic number of the graph are arbitrarily large [59].

2.2.1 Perfectly Orderable Graphs

It is interesting to note that for any graph there does exist some ordering of the vertices such that the greedy colouring algorithm will provide an optimal colouring. This is easy to see, since from any optimal colouring, if we first list out the vertices of colour 1 and then the vertices of colour 2, and so on, a greedy colouring of this order will always produce a colouring with $\chi(G)$ colours. Thus, for general graphs, finding such an order is sufficient to find an optimal colouring, and so the problem of finding this vertex order is NP-hard.

In 1984, Vašek Chvátal [6] defined *perfectly orderable graphs* as those graphs for which there exists a vertex ordering such that, for every vertex subset, if that ordering is observed for that subset, then the greedy colouring method optimally colours the associated induced subgraph. Such an ordering, if it exists, is called a *perfect order*. To illustrate the usefulness of such a definition, imagine a colouring problem on a graph, say, a channel assignment problem to a set of cell phone users. Given a perfect order, we can colour the entire graph optimally. Now if any subset of cell phone users turn off their phones, the resulting users (creating an induced subgraph of remaining vertices) can also be coloured greedily using the

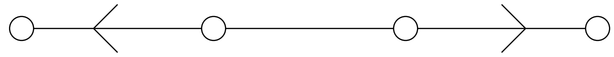

Figure 2.1: An obstruction

already-known perfect order, without having to recompute anything. Two worthy questions are now asked: given a graph, is it perfectly orderable? Given a perfectly orderable graph, how quickly can we find a perfect order?

As an example of a small graph and an ordering which is not perfect, consider a P_4 (v_1, v_2, v_3, v_4) and an ordering v_1, v_4, v_2, v_3. Then the two endpoints v_1 and V_4 receive colour 1 and then v_2 must take colour 2 and v_3 is assigned colour 3. A P_4, however, is easily coloured with two vertices. A simple way to capture this scenario is to orient every edge from v_i to v_j if v_j comes before v_i in the ordering. Then in the above case, since the two endpoints come earliest in the ordering, the wings of the P_4 are oriented outward, while the middle edge is oriented arbitrarily (see Figure 2.1.) When a P_4 is oriented as in Figure 2.1, we call it an *obstruction*.

For graphs with oriented edges, we call an *directed cycle* a sequence of vertices $v_1, \ldots, v_k$ where $\{v_i, v_{i+1}\}$ is an edge oriented from v_i to v_{i+1} and $\{v_k, v_1\}$ is an edge oriented from v_k to v_1. If a directed graph has no directed cycle, we call it *acyclic*.

For an orientation of a graph to represent a perfect order, it must be acyclic (so that a corresponding linear order exists) and it can not have such an obstruction.

Chvátal showed that these necessary conditions are also sufficient.

Theorem 2.2.1 *[6] A graph is a perfectly orderable graph if and only if there exists an acyclic obstruction-free edge orientation.*

The above-mentioned questions were answered by Middendorf and Pfeiffer in 1986 [47] when they proved that perfectly orderable graph recognition is NP-Complete. It follows that finding a perfect order (even in a graph known to be perfectly orderable) is NP-hard, since if we had an algorithm to find a perfect order we could apply it to any graph, construct the associated acyclic orientation and check if it is valid simply by checking the wings of every P_4 to ensure all P_4s are obstruction-free. The validity of the orientation would then answer the question of whether the original graph is perfectly orderable.

Many subclasses of perfectly orderable graphs have been introduced. *Brittle* graphs [7] are perfectly orderable and polynomially recognizable, but no forbidden induced subgraph characterization is known for them. A subclass of brittle graphs which have are characterizable by forbidden subgraph is the class of *(house, hole, domino)-free* graphs [34] (HHD-free graphs). These two classes are described in subsection 3.2.4, as HHD-free graphs are an important class in relation to the $(P_5, \overline{P}_5, C_5)$-free graphs.

2.2.2 Optimizing with a Perfect Order

Chvátal [6] showed that perfectly orderable graphs are perfect; that is, he showed that for every induced subgraph of a perfectly orderable graph, the maximum clique size is equal to the chromatic number. Since the definition of a perfect order provides an optimal colouring in linear time when a perfect order is known, the fact that these graphs are perfect also gives the maximum clique size as well.

To find a maximum clique in a perfectly orderable graph G, consider a perfect order $<$ on the vertices. Consider a vertex v of largest colour, and due to the colouring scheme, we know it must have a neighbour in each colour class corresponding to colours smaller than $\chi(G)$. We can initialize the clique with two vertices, v and one of its neighbours of the next highest colour, call it w. Chvátal proved the following lemma:

Lemma 2.2.2 *[6] Let G be a graph with a perfect order $<$ and a clique C with every w in C having a neighbour $p(w)$ not in C. Let $P = \{p(w)$, for every $w \in C\}$. If P forms a stable set, then some $p(w)$ in P is adjacent to all of C.*

Here, we have a $C = \{v, w\}$, and choose the neighbours $p(v)$ and $p(w)$ to be in the next lowest colour class so that we know they have no edge between them. The proof of the lemma is algorithmic, as so provides us with the desired $p(w)$ vertex adjacent to all of C. The clique is extended to include this vertex, and the process is repeated.

The problem of finding a maximum independent set in a perfectly orderable graph can be solved in polynomial time by virtue of the fact that perfectly orderable graphs are perfect. However, as we discussed earlier, the algorithms on perfect graphs are not very practical. The problem of finding a maximum independent set of a perfectly orderable graph with a polynomial-time combinatorial algorithm is still open, with or without a given perfect order. In fact, it is still unknown as to how one could produce an optimal colouring and maximum clique in a perfectly orderable graph with a purely combinatorial method in polynomial time when a perfect order is not provided.

2.3 Modular Decomposition

A common problem-solving technique is to break a large problem into smaller subproblems, usually in hopes of making the subproblems simple enough to solve and whose solutions can be somehow combined to solve the original, large problem. This section discusses a method for decomposing graphs using a method called *modular decomposition*. When modular decomposition is applied to cographs, it is exactly the same as cograph decomposition already discussed (in Subsection 1.3.2,) and so it can be considered a kind of generalization of the cograph decomposition.

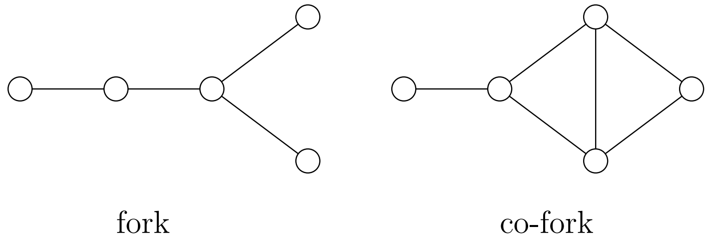

Figure 2.2: The fork and its complement

When a graph is no longer decomposable with respect to successive graph complementations or modular decomposition, it is called *prime*. The prime cographs are single vertices. Later, we will see how the modular decomposition has been used to recognize (P_5, $\overline{P}_5$, bull)-free graphs, whose prime graphs are simply bipartite or complements of bipartite graphs. The prime semi-P_4-sparse are either bipartite graphs, complements of bipartite graphs, or a special kind of split graph.

2.3.1 Homogeneous Sets

Cographs were introduced in Subsection 1.3.2. Recall that cographs are those graphs totally decomposable (to single vertices) by successively taking complements of connected components. A *co-connected component* (or *co-component*) is a set of vertices which form a connected component in the complement of the graph. Hence, the cograph decomposition breaks the graph into its components and then each component into co-components, and so on. Observe that a component C_1 has the property that every vertex in $V - C_1$ sees none of C_1, and similarly a co-component C_2 has the property that every vertex in $V - C_2$ sees every vertex in C_2. We say that with respect to a graph, a vertex subset S is *indistinguishable* if every vertex outside the set either sees every vertex in S or misses every vertex in S. Indistinguishable sets have also been called *homogeneous sets* or *modules*.

Definition 2.3.1 *$S \subseteq V$ is a* module *if $1 < |S| < |V|$, and every $v \in V - S$ either sees every $s \in S$ or misses every $s \in S$.*

Definition 2.3.2 *A graph is* prime *if it has four or more vertices and contains no module.*

An example of a graph which is connected and co-connected (and so can not be decomposed in the way that cographs can be) while still having a module is the *fork*, whose graph and complement are depicted in Figure 2.2. If a set $M \subseteq V$ is a module of G, then that set is also a module in $\overline{G}$ since the sets of vertices in $V - M$ that miss all of M and see all of M switch roles.

Note some basic properties of modules:

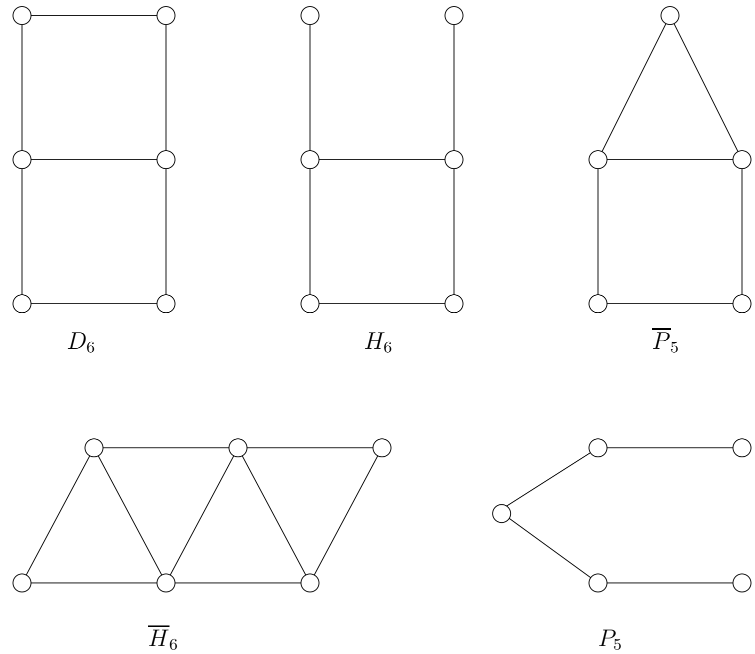

Figure 2.3: Various graphs

Proposition 2.3.3 *[3] Let $G = (V, E)$ be a graph and let A, B be two modules of G. Then the following properties hold:*
(i) $A \cap B$ is a module
(ii) if $B \not\subseteq A$, then $A \setminus B$ is a module
(iii) if $A \cap B \neq \emptyset$, then $A \cup B$ is a module.

These properties allow us to have a well-defined notion of *maximal modules*, since if two modules intersect, either one is completely contained in another or the union of the two modules form a larger module.

2.3.2 Prime Graphs

Recall that graphs with no modules are called prime. Any graph with more than one vertex that has no P_4 (and hence is a cograph) will have a module (a component or a co-component,) and so any prime graph must have a P_4. In fact, the only prime graph on four or fewer vertices is the P_4. The prime graphs on five vertices are the *bull*, P_5, $\overline{P}_5$, C_5 (see figure 2.3).

It was earlier stated that modules are invariant under complementation, so a graph G is prime if and only if $\overline{G}$ is prime. More prime graphs include the $P_{k>6}$, $\overline{P}_{k>6}$, $C_{k>6}$, $\overline{C}_{k>6}$.

Even though a graph such as C_4 is not prime itself, the definition of primality does not restrict C_4 from existing in a prime graph. The definition of primality simply implies that the rest of the graph must adjoin to the C_4 so that the modules in the C_4 are not modules in the entire graph. This suggests the existence of theorems of the sort *if a graph is prime*

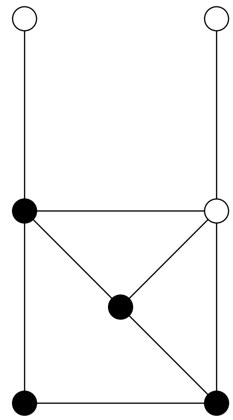

Figure 2.4: A prime $(P_5, \overline{P}_5)$-free graph with a C_4 (filled) that is not in an H_6.

and has a certain non-prime subgraph, then that non-prime subgraph must extend in some way to some prime structure. One of the first theorems of this sort is given by Hoàng and Reed:

Theorem 2.3.4 *[35] If a prime graph has an induced C_4 then the graph must have at least one of a D_6, H_6 or a $\overline{P}_5$ (Figure 2.3).*

Graphs such as D_6, H_6 and $\overline{P}_5$ are called *minimal prime extensions* of the graph C_4, since any prime graph containing a C_4 must contain one of these. The minimal prime extensions of small graphs (those one five or fewer vertices) have been completely characterized; some graphs have an infinite number of minimal prime extensions [2].

It should be noted here that the term "prime extension" may be misleading. Let S be set of four vertices inducing a C_4 in a prime graph G. Theorem 2.3.4 tells us that G must contain six vertices which induce one of D_6 or H_6 or else has five vertices inducing $\overline{P}_5$. Let T be a set of five or six vertices in G inducing one of the minimal prime extension of S. The reason "extension" may be misleading is because it is not necessarily the case that S is a subset of T. Figure 2.4 shows such an example.)

In a graph G with a module M, we can *shrink* M to a single vertex v_M such that v_M sees the vertices in G that were universal on the module M and sees no other vertices. This leaves behind the graph induced by the vertex set $V \setminus (M \setminus v_M)$, where v_M can be taken as any vertex in M. When each maximal modules of G has been replaced by a single vertex, we call the resulting graph G^* the *characteristic graph of* G. Note that G^* is prime and is an induced subgraph of G.

2.3.3 Modular Decomposition and Partial Closure

When a module M of a graph G has been replaced with a single vertex, the graph G is said to have been *decomposed* into the following two graphs: the graph induced by $V \setminus (M \setminus v_M)$

and the graph induced by M, and these two graphs can be decomposed further. This procedure of successively decomposing a graph is the *modular decomposition*. The linear time recognition algorithm for decomposing cographs by Corneil, Perl and Stewart [11] led to the development of a linear time algorithm for constructing the modular decomposition tree (by McConnell and Spinrad [44] and by Cournier and Habib [12].)

The general linear time decomposition algorithm is difficult and not discussed here. A simpler linear time algorithm for modular decomposition exists for various classes, for example, chordal graphs [37].

For computer-aided searches during the research for this thesis, a modular decomposition was not used. We found all prime graphs on ten or fewer vertices using a simple recognition algorithm. For a set of vertices S in G, if v is a vertex in $V(G) - S$ we call v *partial* on S if v sees a vertex in S and misses a vertex in S. Following [31], for any vertex subset S of V with $|S| > 1$, define $PTL(S)$ to be the set of vertices in S together with any vertices in $V - S$ which are partial on the set S. As is usual with functional notation, we let $PTL^1(S) = PTL(S)$ and $PTL^i(S) = PTL(PTL^{i-1}(S))$, for $i > 1$. Since $|PTL(S)| \geq |S|$, we are guaranteed to have some k_S for which $PTL^{k_S}(S) = PTL^{k_S-1}(S)$. Define the *partial closure of* S, $PTL^*(S)$, as the set of vertices obtained after iterating the $PTL(S)$ function k_S times. In particular, notice that if S is homogeneous in G, then $PTL(S) = S$.

A simple (while not necessarily efficient) algorithm to determine whether a graph is prime relies on the fact that if there exists a module M in a given graph, and if S is some set of vertices with $S \subseteq M$ and $|S| > 1$, then $PTL^*(S) = M_2$ where M_2 is some module such that $M_2 \subseteq M$ with possibly $M_2 = M$. It follows that the partial closure just needs to be tested for all S with $|S| = 2$, and if any such partial closure is a strict subset of the whole graph then we have found a module, otherwise the graph is prime.

2.4 Relevant Properties of Prime Graphs

There are some important properties of prime graphs to note. For any pair of vertices $\{x, y\}$ in a proper vertex subset S of V in a prime graph, there must be some vertex d_1 that sees x and not y, or sees y and not x. However, since the three vertices $\{d_1, x, y\}$ also cannot form a module, there must be another vertex d_2 partial on this set, and by continuing this process we do not explicitly know which d_i vertices are inside or outside the set S (even though we know S is not a module and there must exist a vertex outside S which is partial on it, possibly seeing multiple vertices and missing multiple vertices of S.) For specific local information, the following lemma is indispensable. Call a set *big* if it has size at least two.

Lemma 2.4.1 *[19] Let G be a prime graph and S a strict subset of $V(G)$ with $|S| > 1$. Let the big connected components of S be $Q_1, Q_2, \ldots, Q_k$, (respectively, co-connected components*

$R_1, R_2, \ldots, R_l$.) Then for any $1 \leq i \leq k$ ($1 \leq j \leq k$) there exists some vertex in $V(G) \setminus S$ which is partial on some edge in Q_i (non-edge in R_j).

Recall that a vertex in a graph is called *simplicial* if its neighbourhood induces a clique. Call a vertex *co-simplicial* if the vertex is simplicial in the graph's complement. That is, a vertex is co-simplicial if its non-neighbours induce a stable set.

Lemma 2.4.2 *The set of simplicial vertices in a prime graph induces a stable set.*

Proof. Let v and w be two simplicial vertices in a prime graph. There must exist some d distinguishing $\{v, w\}$, so say d sees v and misses w. If v sees w then v sees both d and w, but d was chosen to miss w which is contrary to v being simplicial. Thus v must miss w. □

Applying the above argument to the graph complement, we have:

Corollary 2.4.3 *The set of co-simplicial vertices of a prime graph induces a clique.*

Chapter 3

Motivating (P_5, $\overline{P}_5$)-free Graphs

The purpose of this chapter is to reveal the connection between (P_5, $\overline{P}_5$)-free or (P_5, $\overline{P}_5$, C_5)-free graphs and other other well-known classes. Later, we will see theorem 4.1.1 of Fouquet telling us that, with respect to prime graphs on six or more vertices, the (P_5, $\overline{P}_5$)-free and (P_5, $\overline{P}_5$, C_5)-free graphs are identical.

3.1 (P_5, $\overline{P}_5$)-free as a Generalization of Subclasses

A natural use of generalization is to take known results on certain graph classes and try to extend those results to a larger class, thereby solving a larger problem. Here, we present some important subclasses which are contained in the class of (P_5, $\overline{P}_5$, C_5)-free graphs.

3.1.1 Split Graphs

A graph is bipartite if its vertices are partitionable into two stable sets; equivalently, a graph is bipartite if it can be coloured with two colours. By changing the vertex partition property from two stable sets to one stable set and one clique, we arrive at the *split graphs*.

Definition 3.1.1 *A graph is* split *if its vertices can be partitioned into two sets S and C, where S is empty or induces a stable set and C is empty or induces a clique.*

Note that the complement of a split graph is a split graph, as the stable set S and clique set C switch roles in the complement. In fact, the class of split graphs is set of graphs which are both chordal and co-chordal.

Theorem 3.1.2 *[16] A graph G is split if and only if G and $\overline{G}$ are chordal.*

Through forbidden induced subgraphs, one may also characterize split graphs as $(C_4, C_5, 2K_2)$-free graphs [16] (where $2K_2$ is a graph on four vertices with two disjoint edges.)

There are many other characterizations of these graphs, including an intersection model (involving the intersections of disjoint stars on a tree, mentioned in [45]) as well as vertex-ordering properties involving degree sequences [27], [58], which lead to a linear time recognition algorithm (in fact, if the degree sequence is given, the algorithm runs in $O(n)$-time.)

Since recognizing split graphs can be performed in linear time, when testing whether a graph is a (P_5, $\overline{P}_5$, C_5)-free graph, we can afford to first test if the graph is split. If it is, having no $2K_2$ ensures that there is no P_5 and having no C_4 tells us that there is no $\overline{P}_5$. The forbidden induced subgraph characterization for split graphs provides the fact that split graphs are a (strict) subclass of (P_5, $\overline{P}_5$, C_5)-free graphs. In the case that testing for split-ness fails, we still gain from our attempt: there are linear time algorithms to recognize split graphs such that if the input graph is not split, a C_4, $2K_2$ or C_5 is found [51]. When testing a graph for containment in (P_5, $\overline{P}_5$, C_5)-free graphs, knowing that it is not split provides us with information on the location of one of these subgraphs.

Yet another benefit from first testing the split-ness of a (prime) graph is that if the graph is not split, we can apply the following theorem:

Theorem 3.1.3 *[31] If a prime graph is not split, then every vertex is in a P_4.*

This theorem appears in the context of P_4-structure [31]. However, this formulation of the theorem aides our study into the structure of prime (P_5, $\overline{P}_5$, C_5)-free graphs.

3.1.2 Recursively Split Substitute Graphs

The operation of *substitution* is the opposite of contracting a module to a single vertex: we replace a vertex v in a graph G with a graph H making every vertex in H adjacent to all the vertices in G that were adjacent to v. In the resulting graph, the subgraph induced by H is a module. For any class of graphs, we can ask whether the substitution of a graph of that class into any vertex of a graph of that class maintains containment within that class. For instance, in the case of split graphs, if we substitute a split graph into another split graph, is the result necessarily a split graph? The answer here is 'no' as we can substitute two nonadjacent vertices into the middle-vertex of a P_3 and obtain a C_4.

The class obtained from substituting split graphs into split graphs is called the *recursively split substitute graphs.* The forbidden induced subgraphs for this class are the $C_5, P_5, \overline{P}_5, H_6$, and the $\overline{H}_6$ (see Figure 2.3) [35] [32].

There is a theorem that characterizes a graph class obtained from substitution:

Theorem 3.1.4 *[61] Let Z be a hereditary class of graphs with forbidden induced subgraphs $\{F_1, F_2, F_3, \ldots\}$ (possibly infinite list) then the class obtained from substituting a Z-graph into a vertex in a Z-graph has forbidden induced subgraphs $MPE(F_1) \cup MPE(F_2) \cup \ldots$, where $MPE(F_i)$ is the set of all minimal prime extensions of F_i [61].*

As a corollary to that theorem, if all the F_i graphs are already prime, the class of graphs is closed under substitution. For instance, $(P_5, \overline{P}_5)$-free and $(P_5, \overline{P}_5, C_5)$-free graphs are closed under substitution.

The *weak bipolarizable graphs* [48] are defined as the *chordal substitute graphs*; that is, a graph is weak bipolarizable if it is obtained from substituting a chordal graph into any chordal graph. The forbidden induced subgraphs of weak bipolarizable graphs are $C_{k\geq 5}$ and all minimal prime extensions of C_4, namely $\overline{P}_5$, H_6, and D_6. Note that these graphs can be characterized as those for which every prime induced subgraph is chordal.

The split substitute graphs can be easily recognized through the fact that a graph is split substitute if and only if it and its complement are both weak bipolarizable. This follows from the forbidden induced subgraph definition of weak bipolarizable graphs by Olariu in [48]. The paper also gives a linear time recognition algorithm for weak bipolarizable graphs. Since $(C_5, P_5, \overline{P}_5, H_6, \overline{H}_6)$-free graphs can be recognized in $o(n^3)$ time, it is natural to wonder if the $(P_5, \overline{P}_5, C_5)$-free graphs can be as well.

3.1.3 Cographs

Much discussion has already been devoted to cographs (P_4-free graphs) and their importance in the study of graph classes and graph algorithms. One might wonder if the simple generalization from P_4-free graphs to P_5-free graphs would yield as fruitful a study as the study of cographs has been. Surprisingly, very little is known about P_5-free graphs. In fact, the max independent set problem is of unknown complexity [21] on this class of graphs, while it is easily polytime solvable on P_4-free graphs (as already described) and it is known to be NP-complete on P_6-free graphs.

The cograph decomposition tree (through graph complements) and its use in solving such problems as max independent set, max clique, min colouring and min clique cover relies heavily on the fact that P_4-free graphs are self-complementary. Since P_5-free graphs do not share this property, the self-complementary class of $(P_5, \overline{P}_5)$-free free graphs may exhibit generalizations of properties of cographs.

3.2 $(P_5, \overline{P}_5, C_5)$-free Graphs as a Special Case of Superclasses

Rather than relaxing restrictions to arrive at more general graph classes, one may impose further restrictions to gain more structure and properties in graphs. We will see here how our $(P_5, \overline{P}_5, C_5)$-free graphs serve as an important subclass of other well-studied classes.

3.2.1 Welsh-Powell Perfect Graphs

Early investigations into graph colouring led to the empirical analysis of heuristic methods to colour graphs. One such heuristic was to sort the vertices of a graph by their degree (from largest to smallest) and then to apply a greedy colouring to this ordering [59]. The intuition here being to colour the potentially most-constrained vertices first. The vertex ordering based on degree is sometimes referred to as a *Welsh-Powell ordering*. In 1987, Chvátal et. al. [9] defined and studied several subclasses of perfectly orderable graphs, one of which they called *Welsh-Powell perfect graphs*:

Definition 3.2.1 *[9] A graph G is* Welsh-Powell perfect *if every Welsh-Powell ordering is a perfect ordering for that graph.*

In [9], a forbidden induced subgraph characterization is given for Welsh-Powell perfect graphs, which includes 17 forbidden subgraphs. One of these is the C_5, and it was observed that every other forbidden subgraph contained a P_5 or its complement, and so the (P_5, $\overline{P}_5$, C_5)-free graphs are Welsh-Powell perfect. Hence, it is easy to find a perfect ordering for (P_5, $\overline{P}_5$, C_5)-free graphs using the degree sequence, and moreover, these graphs are self-complementary. Linear time algorithms were given in [9] to solve clique, colouring, independent set and clique cover on Welsh-Powell perfect graphs, and thus on (P_5, $\overline{P}_5$, C_5)-free graphs.

These (P_5, $\overline{P}_5$, C_5)-free graphs are not the largest self-complementary class of perfectly orderable graphs. Clearly, the largest such class is the class of perfectly orderable graphs intersected with the co-perfectly orderable graphs, but no alternate characterization for this class is known to exist (particularly, no forbidden induced subgraph characterization is known for this.) We will see in Subsection 3.2.4 the self-complementary class of *brittle* graphs that do not include all perfectly orderable co-perfectly orderable graphs, but do include the (P_5, $\overline{P}_5$, C_5)-free graphs as a subclass.

3.2.2 Meyniel Graphs

Recall that a graph G is is perfect if for every induced subgraph H of G the clique size of H is equal to the chromatic number of H. The long-standing Strong Perfect Graph Conjecture ("SPGC" resolved in 2003, [5]) conjectured that a G is perfect if and only if G and $\overline{G}$ had no induced odd cycles of size five or more. (Calling a graph G *Berge* if G contains no odd holes of size five or more, nor complements of odd holes, we can re-state the SPGC to be: G is perfect if and only if G is Berge.)

Note that the statement "no induced odd cycles" is equivalent to saying "every odd cycle has a chord." In the 1970's, Meyniel proved a theorem weaker than the SPGC:

Theorem 3.2.2 *[46] Let G be a graph where every odd cycle has at least two chords. Then G is perfect.*

Such graphs are now called *Meyniel Graphs*. Meyniel graphs have been studied for their recognition problem extensively, and graph tools such as the amalgam (not described here, see [4]) have been developed as a result of investigations into Meyniel graphs. Recognition algorithms for Meyniel graphs have often been useful in the recognition of other graph classes such as *i-triangulated graphs* [52] and *quasi-Meyniel graphs* [15]. Meyniel graph recognition has been improved to $O(m^2 + mn)$ by Roussel and Rusu [52].

The forbidden induced subgraphs for Meyniel graphs are odd holes and odd cycles with exactly one chord. In particular, C_5 and $\overline{P}_5$ are two of the forbidden subgraphs. The remainder of the forbidden graphs – the next smallest of which are the 7-cycle and the two ways that a 7-cycle can have a single chord – contain a P_5.

Theorem 3.2.3 *A graph G is a P_5-free Meyniel graph if and only if it is a (P_5, $\overline{P}_5$, C_5)-free graph.*

Using the fact that Meyniel graphs are $\overline{P}_5$-free, we see that co-Meyniel graphs are P_5-free, and since the C_5 is self-complementary, the largest self-complementary class of Meyniel graphs (the Meyniel co-Meyniel graphs) are exactly the (P_5, $\overline{P}_5$, C_5)-free graphs [3].

Theorem 3.2.4 *[3] A graph G is (P_5, $\overline{P}_5$, C_5)-free if and only if G and $\overline{G}$ are Meyniel.*

3.2.3 Weakly Chordal and Murky Graphs

Meyniel graphs were a natural special class of Berge graphs and were shown to be perfect. Another natural special class of Berge graphs can be formed, not by forcing more chords in the odd cycles, but instead by forbidding all large cycles rather than just the odd ones.

Definition 3.2.5 *[28] A graph G is* weakly chordal *if neither G or $\overline{G}$ has a chordless cycle of size five or more.*

Recalling that chordal graphs are graphs with no induced cycles of size four or more, and noting that the complements of odd holes of size six or larger must have an induced subgraph isomorphic to a C_4, it follows that weakly chordal graphs contain both chordal graphs and complements of chordal graphs.

As chordal and co-chordal graphs are known to be perfect [23], the generalization to weakly chordal graphs as a step closer to the SPGC as weakly chordal graphs are Berge graphs. In 1985, Hayward showed the following:

Theorem 3.2.6 *[28] Weakly chordal graphs are perfect.*

Weakly chordal graphs have come to be known as a large class of graphs containing many other well-studied classes such as brittle graphs (see Section 3.2.4,) HHD-free graphs (Section 3.2.4,) HH-free graphs (Section 3.2.5,) domination graphs (Section 3.2.5,) (P_5, $\overline{P}_5$, C_5)-

free graphs, and many others not mentioned in this thesis. Recently, weakly chordal graphs have found themselves used in the field of bioinformatics [1] [43].

In continuing the class-broadening towards Berge graphs from weakly chordal graphs, Hayward introduced *murky graphs*.

Definition 3.2.7 *[29] A graph is* murky *if it contains no C_5, P_6, or $\overline{P}_6$.*

Murky graphs were also shown to be perfect [29], but have received little attention since then. Taking these definitions a step further would call for restricting the C_5, C_7, $\overline{C}_7$, P_8 and $\overline{P}_8$, which is a class which has not been studied to the best of the author's knowledge.

In the same way that murky graphs are a subclass of Berge graphs by the use of a finite number of forbidden subgraphs, the (P_5, $\overline{P}_5$, C_5)-free graphs are a subclass of weakly chordal graphs, as the P_5 and the $\overline{P}_5$ are the common subgraphs to the holes and antiholes of size six and larger. Perhaps an improved recognition of (P_5, $\overline{P}_5$, C_5)-free graphs could be useful in recognizing weakly chordal graphs, which can be currently recognized in $O(m^2)$ [57] or $O(n^4)$-time [30].

3.2.4 Brittle and HHD-free Graphs

After introducing perfectly orderable graphs [6], Chvátal introduced a subclass of perfectly orderable graphs called the *brittle graphs* [7].

Definition 3.2.8 *A graph G is* brittle *if for every induced subgraph H of G either there exists a vertex that is not the midpoint of any P_4 in H or there exists a vertex that is not the endpoint of any P_4 in H.*

Since mid-P_4 vertices and end-P_4 vertices swap roles under graph complementation, brittle graphs are self-complementary. To see that brittle graphs are perfectly orderable, note that for an ordering to be obstruction-free, it suffices to colour – for each P_4 – the two mid-vertices before colouring the two end-vertices. So, if a graph is brittle and there is a vertex which is not the end of some P_4, then we can safely put that vertex at the start of a vertex ordering knowing that it will not lead to any obstruction. Similarly (as is seen through complementation) if there is a vertex that is not the mid of some P_4, then we can put that vertex at the end of a vertex ordering. Once one such vertex is placed in a vertex ordering, the remaining graph (without the already-placed vertex) can be considered, and since it is an induced subgraph of the original graph, the definition of brittle graphs tells

[1] A special case of the bioinformatics problem "Single Nucleotide Polymorphism Haplotyping" was reduced to finding a maximum independent set in a class of graphs, and the paper proved their class of graphs were weakly chordal, which allowed for the use of the $O(n^3)$-time algorithms for maximum independent set on weakly chordal graphs. The authors overlooked the fact that they had, in fact, proven their graphs were weakly chordal co-comparability graphs on which there exist linear-time algorithms for maximum independent set. Weakly chordal co-comparability graphs have been studied in their own right, see [14].

us that there is another vertex which can be properly placed into this ordering. Continuing this process until all vertices have been placed yields an obstruction-free ordering. This proves that brittle graphs are perfectly orderable.

The ordering described is more specific than a perfect order and does not necessarily exist for all perfectly orderable graphs, so sometimes this ordering is referred to as a *brittle ordering* [3].

Brittle graphs are difficult to study for their general structure. For instance, no complete characterization through forbidden induced subgraphs is known. Clearly, an induced cycle of size five or more is not brittle since every vertex is a mid-point and an end-point of a P_4, so brittle graphs are hole-free. In [34], Hoàng and Khouzam show that HHD-free graphs are brittle graphs.

Theorem 3.2.9 *[34] Graphs with no $\overline{P}_5$, D_6, and $C_{k\geq 5}$ are brittle.*

These graphs have come to be known as *house, hole, domino free* graphs, or HHD-free graphs. Using the forbidden subgraph characterization, HHD-free graphs can be recognized in $O(n^6)$ time. It is not difficult to describe a faster, and perhaps just as simple, algorithm to recognize these graphs [50]. For every edge $\{x, y\}$ find N_{xy}, the common neighbourhood of x and y, N_x, the neighbours of x and not y, and N_y, the neighbours of y and not x. Let $M = V(G) - N_{x,y} - N_x - N_y - \{x, y\}$. To test whether the edge $\{x, y\}$ is the bottom of a house or the bottom of a domino, perform a path-searching algorithm (say, BFS) from the set of vertices in N_x, through the set M, to the set of vertices in N_y. If a path is found from some s in N_x to some t in N_y using only one vertex from M, then the vertices on the s to t path, together with $\{x, y\}$ form a house if s and t are adjacent or a C_5 if they are not. If two vertices from M are in the s to t path, then a domino is found if s sees t, or a C_6 is s and t are not adjacent. If more than two vertices from M are in the found s to t path, then a large hole is found regardless of whether s sees t.

Building the sets N_{xy}, N_x, N_y, and M and performing path-searching can be implemented to run in $O(m + n)$-time. This process may need to be repeated for every choice of $\{x, y\}$, so this algorithm runs in $O(m^2 + mn)$, or $O(m^2)$ time, an improvement on the brute-force $O(n^6)$-time algorithm.

The recognition of HHD-free graphs has been improved to $O(n^3)$-time by Hoàng and Sritharan [36] using an algorithm that decides whether a given simplicial vertex is the top of a house. In general, HHD-free graphs do not necessarily have simplicial vertices, but they make use of the following theorem:

Theorem 3.2.10 *[34] Every HHD-free graph has a simplicial vertex or a homogeneous set.*

By restricting the consideration of HHD-free graphs to prime HHD-free graphs, Hoàng and Sritharan are able to find simplicial vertices. Since the only way a simplicial vertex can be in a house, hole, or domino is if the vertex is a top of a house, this is the only condition that need be checked. Verifying that a simplicial vertex is not the top of a house allows for the removal of the vertex without affecting the graph's containment in the class of HHD-free graphs, thus reducing the recognition problem to than on a smaller graph on which a modular decomposition can again be applied and the process of checking simplicial vertices repeated.

The only benefit of considering the HHD-free graphs with no modules is to guarantee the existence of a simplicial vertex. The authors of that algorithm comment that a significant improvement will likely utilize new information about the structure of prime HHD-free graphs. This was the first motivation for the author of this thesis to study the class of (P_5, $\overline{P}_5$, C_5)-free graphs, as these are the largest self-complementary class of HHD-free graphs.

3.2.5 Domination and HH-free Graphs

Recall that a graph is chordal if and only if it has a perfect elimination ordering, namely, an ordering of the vertices $v_1, v_2, \ldots, v_n$ where v_j is simplicial in the graph induced by $v_1, \ldots, v_{j-1}$ for every j. By generalising the notion of a simplicial vertex and maintaining the elimination ordering property, a new class of graphs will be described. Firstly, we note that if v is a simplicial vertex then each neighbour u of v sees all of $N(v)$. This motivates the definition of *domination* in graphs:

Definition 3.2.11 *[13] Let u and v be vertices in a graph. Then u* dominates *v if $N(v) \subseteq N(u) \cup \{u\}$. Furthermore, u* strictly dominates *v if $N(v)$ is a strict subset of $N(u)$.*

For notational purposes, u *dominates* v can be represented by $v < u$. We say that u *is comparable to* v if either $u < v$ or $v < u$. Note that domination is self-complementary in that if $u < v$ in G, $v < u$ in $\overline{G}$. If there are vertices u, v such that $u < v$ and $v < u$, then $\{u, v\}$ forms a nontrivial module. Thus, in a prime graph, comparable vertices will always come from strict domination.

Akin to the idea of a perfect elimination ordering, a *domination elimination ordering* (or *d.e.o.*) is an ordering $v_1, \ldots, v_n$ such that every v_j is dominated by some v_i with $i < j$. This leads to the class of *domination graphs*:

Definition 3.2.12 *[13] A graph is a* domination graph *if it has a domination elimination ordering.*

Again, analogous to the characterization of chordal graphs which states that a graph is chordal if and only if every induced subgraph has a simplicial vertex, domination graphs can be characterized as those graphs which have a pair of comparable vertices in every induced subgraph. The inherent properties of vertex domination provide the fact that domination graphs are self-complementary. Domination graphs do not share all the convenient properties of the chordal graphs; for instance, while chordal graphs can be recognized in linear time [53], there are no known polynomial-time recognition algorithms for domination graphs, and it is unknown whether this recognition problem is polytime or NP-complete.

In a C_4 the nonadjacent vertices dominate each other, but in chordless cycles of size five or more there are no comparable vertices. Hence domination graphs can not contain chordless cycles of size five or more. Since domination graphs are self-complementary, it follows that domination graphs are a subset of weakly chordal graphs. This subset relation was shown to be strict when a weakly chordal graph with no comparable vertices on 24 vertices was found by Hayward [28]. It is unknown if this graph is the smallest such example. Though it would seem that domination graphs and weakly chordal graphs form similar classes considering the size of the smallest known graph distinguishing them, Rusu and Spinrad [54] show this is not the case when large graphs are considered. They constructed infinitely many weakly chordal graphs which are minimal non-domination graphs.

Recall that it is NP-complete to recognize perfectly orderable graphs [47]. As a result, HHD-free graphs were developed as they form class having the properties of perfectly orderable graphs while being easier to recognize. Similarly, domination graphs are not yet easily recognized, so the class of *house, hole-free graphs* (HH-free graphs) were introduced in [13] and shown to be domination graphs. Furthermore, the authors showed that every MCS ordering (that is, the breadth-first search ordering where ties between vertices are broken by choosing a vertex adjacent to the largest numbers of vertices already visited) on any HH-free graph is a d.e.o.

As HH-free graphs have been studied in their own right, this provides more motivation to study the (P_5, $\overline{P}_5$, C_5)-free graphs since they are exactly the class of graphs G for which G and $\overline{G}$ are HH-free.

To summarize this chapter, (P_5, $\overline{P}_5$, C_5)-free graphs are exactly the class $\Sigma \cap \overline{\Sigma}$, and also P_5-free Σ, where Σ is any of the well-studied classes of Meyniel graphs, HHD-free graphs or HH-free graphs. (P_5, $\overline{P}_5$, C_5)-free graphs also form a natural subclass of the Welsh-Powell perfect graphs and a superclass of other well-known classes. In addition to all this, (P_5, $\overline{P}_5$, C_5)-free graphs are interesting in their own right, as they can be optimized in linear time, though it takes significantly longer to recognize when a graph is in this class.

Chapter 4

Previous Work on $(P_5, \overline{P}_5)$-free Graphs

In this chapter we recite the results of three different papers on classes that are either subclasses of – or closely related to – the class of $(P_5, \overline{P}_5)$-free graphs.

4.1 $(P_5, \overline{P}_5, \text{bull})$-free Graphs

In a paper titled *"A decomposition for a class of $(P_5, \overline{P}_5)$-free graphs"*, J.L. Fouquet [17] studies the class of $(P_5, \overline{P}_5, \text{bull})$-free graphs. As the bull and P_5 are prime, it is sufficient to show how to recognize prime $(P_5, \overline{P}_5, \text{bull})$-free graphs (that is, prime with respect to the homogeneous set, or modular, decomposition.) Before attacking the problem, a relationship between $(P_5, \overline{P}_5)$-free and $(P_5, \overline{P}_5, C_5)$-free graphs was given:

Theorem 4.1.1 *[17] For each $(P_5, \overline{P}_5)$-free graph G at least one of the following holds:*

i) G has a homogeneous set

ii) G is isomorphic to C_5

iii) G is C_5-free

This theorem implies that prime $(P_5, \overline{P}_5)$-free graphs are exactly the prime $(P_5, \overline{P}_5, C_5)$-free graphs on six or more vertices. The task of characterizing the prime $(P_5, \overline{P}_5, \text{bull})$-free graphs was completed with a fairly elegant solution resulting in a simple structure of these graphs.

Theorem 4.1.2 *[17] Let G be a prime $(P_5, \overline{P}_5, \text{bull})$-free graph on 6 or more vertices. Then G is either bipartite or the complement of a bipartite graph.*

It is not hard to see that every bipartite graph is bull-free and $\overline{P}_5$-free, so the $(P_5, \overline{P}_5, \text{bull})$-free graphs which are bipartite are exactly the P_5-free bipartite graphs. Recognizing prime

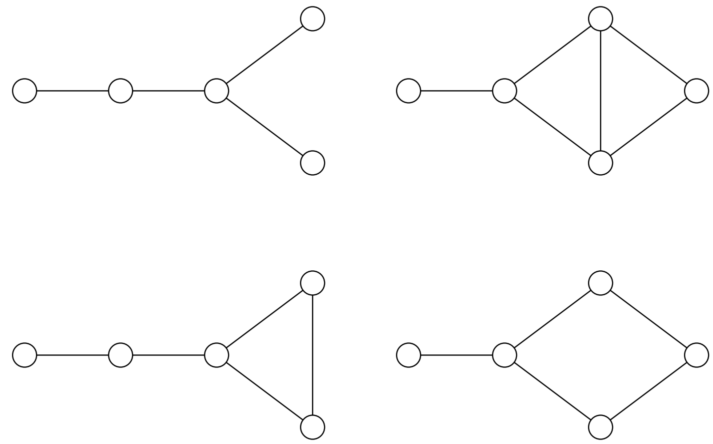

Figure 4.1: The other forbidden induced subgraphs of P_4-sparse graphs

(P_5, $\overline{P}_5$, bull)-free graphs thus reduces to recognizing P_5-free bipartite graphs. Fouquet's analysis of the recognition algorithm lead to an $O(n^3)$ runtime, but this result predates the linear time modular decomposition algorithms of [12] or [44], and so (P_5, $\overline{P}_5$, bull)-free can now be recognized in linear time [18].

4.2 Semi-P_4-Sparse Graphs

As cographs are an important class in the study of algorithmic graph theory, many generalizations to this class have been studied which exhibit similar properties. P_4-sparse graphs were introduced in [33] as a natural generalization of cographs and for which a forbidden induced subgraph characterization is easily obtainable. The class is algorithmically significant because of its linear time recognition through another decomposition scheme, generalizing the standard cograph decomposition.

Definition 4.2.1 *[33] A graph G is* P_4-sparse *if every set of five vertices in G induces at most one P_4.*

Since a set inducing a P_4 in G will also induce a P_4 in $\overline{G}$, this class is self-complementary. In the C_5, as an example, every set of four vertices induces a P_4, so the C_5 is a minimal forbidden induced subgraph of P_4-sparse graphs. Similarly, it is easily seen that the P_5 and $\overline{P}_5$ are minimal forbidden graphs, and so P_4-sparse graphs form a subset of (P_5, $\overline{P}_5$, C_5)-free graphs. The other forbidden induced subgraphs for P_4-sparse graphs are shown in Figure 4.1.

In 1997, Fouquet and Giakoumakis introduced a broader class called the *semi-P_4-sparse* graphs. This class generalizes the P_4-sparse graphs through a relaxation of the forbidden induced subgraphs. They continue restricting the P_5 and $\overline{P}_5$, and only additionally restrict the *kite* graph (which is the complement of the *fork*. See Figure 4.1.) Through a modular decomposition, and using theorem 4.1.1, nontrivial semi-P_4-sparse graphs are also C_5-free.

The structural theorem is almost as simple as the $(P_5, \overline{P}_5, \text{bull})$-free case; however, we require the definition of another graph form.

Definition 4.2.2 *[19] A* thin spider *is a graph whose vertex set can be decomposed into a clique, K, and a stable set, S, such that $|K| = |S|$ or $|K| = |S| + 1$, and the edges between S and K form a matching, leaving at most one vertex of K unsaturated. If there is an unsaturated vertex in K, it is called the* head *of the thin spider.*

The following structural theorem for prime semi-P_4-sparse graphs is simple, yet not sufficient for recognizing the class of graphs.

Theorem 4.2.3 *[19] Let G be a semi-P_4-sparse graph. Then one of the following holds:*

i) G has a homogeneous set

ii) G is bipartite

iii) $\overline{G}$ is bipartite

iv) G is a thin spider

v) $\overline{G}$ is a thin spider

In the $(P_5, \overline{P}_5, \text{bull})$-free case, recognition was accomplished by the fact that the forbidden graphs were prime, and so the original graph before decomposing would be free of those configurations if the graphs obtained from the modular decomposition were also free of them. Since the kite is not a prime graph, the semi-P_4-sparse graphs do not share the convenience of only verifying the primality of the leaves in the modular decomposition tree. The kite has two vertices forming a homogeneous set which could potentially vanish under modular decomposition if the rest of the graph around them does not distinguish them. Fouquet and Giakoumakis show that it is easy to detect if this has indeed happened using a trick of marking certain vertices during the decomposition scheme. During the decomposition, when a module is removed and replaced by a single vertex, if that module was not a stable set then the vertex replacing that module will be *marked*. This implies that if, in a resulting prime graph, there is a P_4 having a middle vertex marked, then that middle vertex must have replaced some edge, and with the rest of the P_4 it would create a kite. Hence the ends of P_4 may be marked without any consequence of having kites in the original graph.

The authors fully characterize the allowable positions of marked vertices in the resulting bipartite graph or thin spider, and show that those facts lead to a recognition algorithm in linear time. Whether a module has an edge can be determined in linear time if certain data structures are used [56].

A *4-pan* (sometimes called the P graph) is the graph formed by substituting two non-adjacent vertices for a middle vertex in a P_4. The 4-pan can be thought of as a kite whose two vertices forming a homogeneous set are non-adjacent, rather than adjacent. One might investigate (P_5, $\overline{P}_5$, 4-pan)-free graphs and expect them to be recognized similarly to the semi-P_4-sparse graphs, only with the difference of marking vertices when they replace non-stable modules, instead of non-clique modules. However, the structure of prime (P_5, $\overline{P}_5$, 4-pan)-free graphs is not as simple as the structure of prime semi-P_4-sparse graphs and so recognition does not reduce to simply recognizing P_5-free bipartite graphs or spiders. The resulting prime graphs have not yet been characterized, and recognition would further require a complete characterization of the valid and invalid markings of such prime structures.

There is another characterization for the prime (P_5, $\overline{P}_5$, 4-pan)-free graphs:

Theorem 4.2.4 *The set of prime (P_5, $\overline{P}_5$, 4-pan)-free graphs is equal to the set of prime P_5-free chordal graphs.*

Proof. Clearly, a P_5-free chordal graph is (P_5, $\overline{P}_5$, 4-pan)-free. To see that prime (P_5, $\overline{P}_5$, 4-pan)-free graphs are also prime P_5-free chordal graphs, recall that prime (P_5, $\overline{P}_5$)-free graphs have the property of being C_5-free, and if a prime graph contained a C_4, it must contain either a $\overline{P}_5$ or a 4-pan. □

There is no known characterization of P_5-free chordal graphs which allows linear time recognition, despite the many distinct characterizations of chordal graphs and their associated recognition algorithms.

4.3 (P_5, $\overline{P}_5$)-Sparse Graphs

Recently, Fouquet and Vanherpe [20] began the study of a class of graphs generalizing the (P_5, $\overline{P}_5$)-free graphs in the same way that P_4-sparse graphs generalize cographs.

Definition 4.3.1 *[20] A graph is called* (P_5,$\overline{P}_5$)-sparse *if in every set of six vertices, the number of induced P_5s and induced $\overline{P}_5$s is at most 1.*

This study led to analogues of some theorems on (P_5, $\overline{P}_5$)-free graphs, such as the restriction of the C_5 in prime graphs.

Theorem 4.3.2 *[20] Let G be a prime (P_5, $\overline{P}_5$)-sparse graph. Then G is either isomorphic to a C_5 or is C_5-free.*

After establishing this property, the authors fully characterize the prime (P_5,$\overline{P}_5$,bull)-sparse graphs, leading to a linear-time recognition algorithm.

Chapter 5

Four Other Self-Complementary Classes of Perfectly Orderable Graphs

Since the recognition of perfectly orderable graphs is NP-complete, subclasses of perfectly orderable graphs for which recognition is polynomial time are desirable. Some subclasses have already been mentioned, such as the brittle graphs (see section 3.2.4). Brittle graphs have no known forbidden induced subgraph characterization, but they are known to contain HHD-free graphs [34] (see section 3.2.4 for information on HHD-free graphs.) The intersection of HHD-free graphs and co-HHD-free graphs is the set of (P_5, $\overline{P}_5$, C_5)-free graphs, as mentioned earlier, and self-complementarity is an elegant and useful property to have in a graph class.

We can define several natural classes of self-complementary perfectly orderable graphs:

Definition 5.0.3 *Let $<$ represent a total ordering on the vertices, and $>$ represent the reverse order of $<$. We can define the following four classes.*

PO1: *perfectly orderable $\cap$ co-perfectly orderable*

PO2: *graphs G which admit a perfect ordering $<$ such that at least one of $<$ or $>$ is perfect in $\overline{G}$*

PO3: *graphs G which admit a perfect ordering $<$ such that $<$ is also perfect in $\overline{G}$*

PO4: *graphs G which admit an ordering $<$ such that both $<$ and $>$ are perfect in both G and $\overline{G}$*

Note that PO4 $\subset$ PO3 $\subset$ PO2 $\subset$ PO1. Since brittle graphs are self-complementary and perfectly orderable, we have that brittle graphs are PO1. We can go further, noticing that a brittle ordering in the complement graph is the reverse of a brittle ordering, and so brittle graphs are PO2. It is not clear how brittle graphs relate to the other two classes.

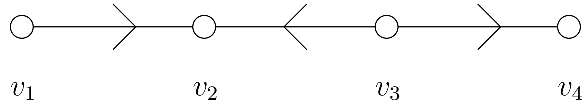

Figure 5.1: An orientation from a PO4 vertex ordering

We note that PO4 is not a subset of HHD-free graphs or (P_5, $\overline{P}_5$, C_5)-free graphs since the house is PO4. All these classes are subsets of weakly chordal perfectly orderable graphs, since perfectly orderable graphs are $\overline{C}_{k\geq 5}$-free. Later, we show a (P_5, $\overline{P}_5$, C_5)-free graph which is not PO4, so the (P_5, $\overline{P}_5$, C_5)-free graphs and PO4 graphs form two sets which are incomparable with respect to set containment.

When a graph has a total ordering on its vertices, there is an associated orientation of its edges which orient edge $\{u, v\}$ from u to v if and only if v comes before u in the vertex order. Recall that perfectly orderable graphs are characterized as those graphs which admit an ordering for which the associated edge-orientation is obstruction-free, where an obstruction is a P_4 (v_1, v_2, v_3, v_4) such that $v_1 < v_2$ and $v_4 < v_3$. Let us decompose this orientation into the possible orderings that would bring about this obstruction. If a P_4 is labeled (v_1, v_2, v_3, v_4), then this P_4 being obstruction-free is equivalent to saying that the ordering on these four vertices cannot be any of

$$\begin{array}{ccccccc}
v_1 & < & v_2 & < & v_4 & < & v_3, \\
v_1 & < & v_4 & < & v_2 & < & v_3, \\
v_1 & < & v_4 & < & v_3 & < & v_2, \\
v_4 & < & v_1 & < & v_2 & < & v_3, \\
v_4 & < & v_1 & < & v_3 & < & v_2, \\
v_4 & < & v_3 & < & v_1 & < & v_2.
\end{array}$$

The edge-orientation groups these six cases into one equivalence class.

Unfortunately, the classes above are not as easily described with the edge-orientation. As an example, if every P_4 (v_1, v_2, v_3, v_4) in a graph is ordered $v_2 < v_4 < v_1 < v_3$ then the graph would belong in class PO4. However, it would not be correct to say that every P_4 can be oriented as in figure 5.1, since an ordering $v_2 < v_4 < v_3 < v_1$ leads to the same orientation but is not a valid P_4 ordering for the class PO4. There is, however, a simple way to characterize the PO4 graphs with a Chvátal-like characterization:

Theorem 5.0.4 *A graph G is in PO4 if and only if G admits an ordering such that every P_4 (v_1, v_2, v_3, v_4) has its vertices visited in the order $v_i < v_j < v_k < v_l$ where $i+l = j+k = 5$.*

Proof. The theorem statement is simply a compact way of saying that every P_4 (v_1, v_2, v_3, v_4) must have its vertices visited in one of the four following ways:

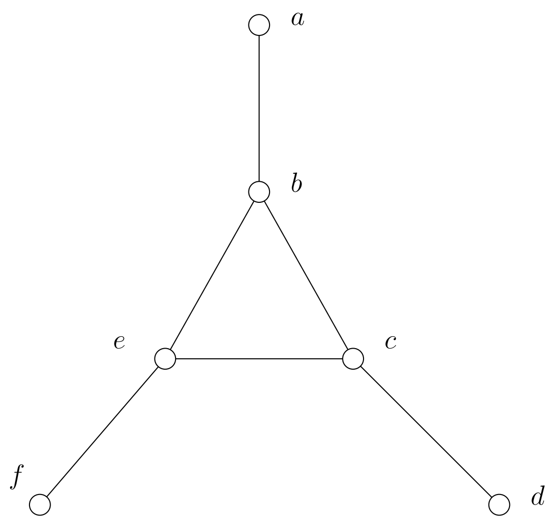

Figure 5.2: The net graph

$$v_1 < v_2 < v_3 < v_4$$
$$v_1 < v_3 < v_2 < v_4$$
$$v_4 < v_2 < v_3 < v_1$$
$$v_4 < v_3 < v_2 < v_1$$

First, we show that if we have such an orientation on G, then G is in PO4. Observe that in the complement graph, the two end-vertices switch roles with the two mid-vertices, so the above property of an ordering is independent of graph complementation, and it is also independent of order reversal. The six forbidden orderings which correspond to an edge-oriented obstruction are given above, and so it is easily verified that an ordering with our given property never creates an obstruction. Thus the ordering and its reverse are perfect for G and $\overline{G}$.

To prove that a PO4 graph must admit such an ordering we can exclude forbidden orders from the $4! = 24$ possible orders on four vertices. Removing the six orderings corresponding to an obstruction, as well as all of their reverse orderings, leaves 12 possible permutations left. Under complementation, a P_4 a, b, c, d maps to b, d, a, c (or the reverse, but we need not be concerned with that since our properties are reversal-independent). The 12 forbidden orderings already removed from consideration will map to 12 more orderings under this morphism, and it is a simple task to identify any new forbidden orderings. These 12 orderings contain four more forbidden orderings, so removing those four as well as their respective reverses leaves behind the four orderings given above, completing the proof. □

Since paths P_k are all in the class PO4 and long holes are not, we have that $C_{k\geq 5}$ and $\overline{C}_{k\geq 5}$ are minimal non-PO4 graphs. Much in the same way that no minimal non perfectly

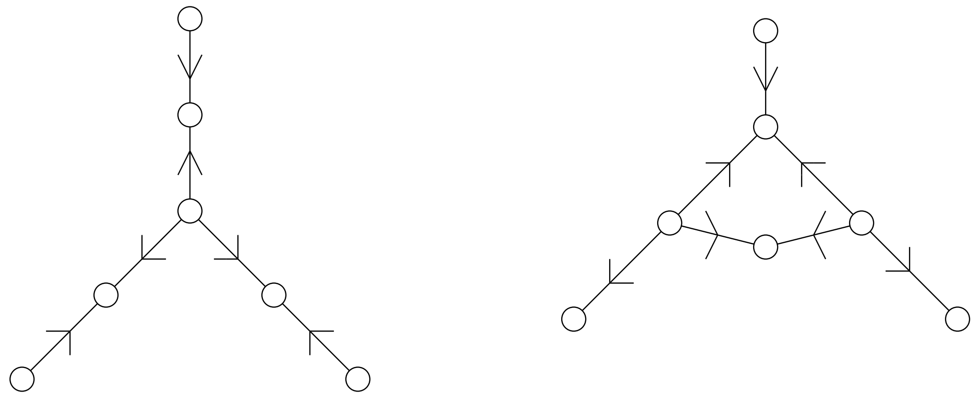

Figure 5.3: Two minimal non-PO4 graphs with one-in-one-out orientations

orderable graph has a homogeneous set [50], the property exists for these graph classes as they are defined in terms of perfect orders. The only other minimal non PO4 graphs on six or fewer vertices are D_6, $\overline{D}_6$, the net and its complement. The net graph is shown in Figure 5.2. Before proving that the net graph is not PO4, we first mention another class of perfectly orderable graphs sometimes referred to as *one-in-one-out* graphs:

Definition 5.0.5 *[50] A graph is* one-in-one-out *if its edges can be oriented so that every P_4 has one wing oriented inward and the other wing oriented outward.*

It is currently unknown whether one-in-one-out graphs can be recognized in polynomial time [50]. We note that every PO4 ordering will have an associated orientation which is one-in-one-out, but not every one-in-one-out orientation gives a PO4 ordering. For instance, the P_4 (v_1, v_2, v_3, v_4) could be ordered $v_1 < v_3 < v_4 < v_2$ but this is not a permissible PO4 ordering. The important property of one-in-one-out graphs we will use here is that if $v_1 < v_2$ then we must have $v_3 < v_4$. The one-in-one-out class was defined as a special case of perfectly orderable graphs along with several other classes by creating special conditions on the allowable orientations of P_4s. Of those classes defined, the recognition complexity has been classified as either polytime or NP-complete for all except for the one-in-one-out graphs.

Proposition 5.0.6 *The net graph is a minimal non-PO4 graph.*

Proof. Consider the net as labeled in Figure 5.2, and assume there is some PO4 ordering on it. Because of the symmetry of the graph, we can assume without any loss of generality that $a < b$. By the property mentioned above, we must have $c < d$ and $e < f$, but then we have the P_4 (d, c, e, f) which does not satisfy the conditions for a PO4 ordering. □

Corollary 5.0.7 *The net graph is a minimal forbidden graph for both PO4 graphs and one-in-one-out graphs.*

The corollary follows since the removal of any one vertex leaves behind only a single P_4 which could be oriented in any way desirable. Since the net graph is split, and thus (P_5, $\overline{P}_5$, C_5)-free and HHD-free, we have that the PO4 class is not comparable to these three mentioned classes. The $\overline{P}_5$ is an example of a graph which is not split, HHD-free, nor (P_5, $\overline{P}_5$, C_5)-free while being in PO4. Graphs on four or fewer vertices are examples of graphs belonging to all of the HHD-free, (P_5, $\overline{P}_5$, C_5)-free , and PO4 graphs.

Two more graphs which are minimal non-PO4 graphs are shown in Figure 5.3, along with edge orientations showing that these are one-in-one-out graphs, unlike the net graph. The details showing these are minimal non PO4 graphs are omitted, but we outline the reasoning which can lead to proof.

An exhaustive search can show that any valid PO4 orderings of a P_5 is the reverse of, or equivalent to, one of the three shown in figure 5.4. Here, "reverse" implies a reading of the P_5s from right to left, and "equivalent" means equivalent under the mapping i to $6 - i$. For example, the reverse of case B) in Figure 5.4 is $4, 5, 2, 3, 1$ while an equivalent ordering would be $5, 3, 4, 1, 2$. Using those three ways to visit the vertices of a P_5, one can add a vertex seeing the middle vertex of the P_5 and find three (again, up to equivalence and reversals) valid orderings of this graph. After adding the final vertex to make either of the two graphs shown, it is seen that no suitable ordering exists to accommodate the final vertex.

To the best of the author's knowledge, the recognition status of the four classes PO1 - PO4 is open.

Since LexBFS serves as a natural method to order vertices in graphs, and many types of orderings that have been studied such as the perfect elimination ordering, semiperfect elimination ordering, domination elimination ordering, perfect ordering mentioned in this thesis so far, there is an associated class of graphs which admit such properties under any LexBFS ordering. For example, we have the following theorems:

Theorem 5.0.8 *[53] Every LexBFS ordering of G is a perfect elimination ordering if and only if G is a chordal graph.*

Theorem 5.0.9 *[38] Every LexBFS ordering of G is a semiperfect elimination ordering if and only if G is HHD-free.*

Theorem 5.0.10 *[13] Let G be a HH-free graph. Then every LexBFS ordering is a domination elimination ordering.*

Since we have defined the PO4 ordering, we are inclined to state the following property:

Theorem 5.0.11 *Every LexBFS ordering of a graph G is a PO4 ordering if and only if G is a cograph.*

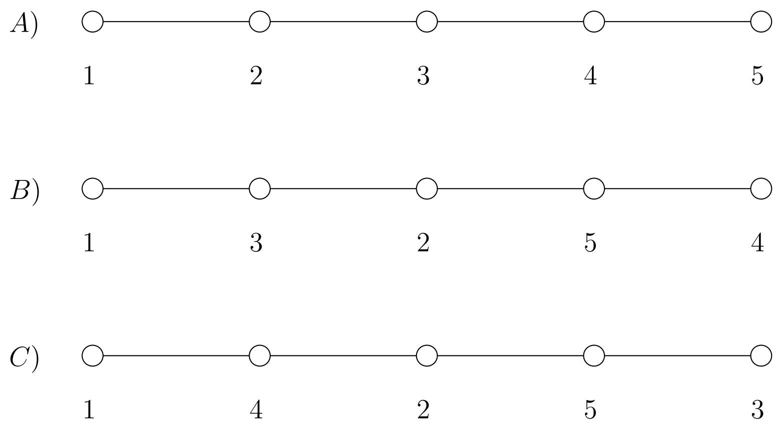

Figure 5.4: Three distinct PO4 orderings of a P_5

Proof. If G is P_4-free, then clearly it is PO4 since there are no P_4s that can violate the characterizing property. If G has a P_4 a, b, c, d then there is a LexBFS ordering that takes $b < c < a < d$ which is not a valid PO4 ordering. □

This BFS ordering on the P_4 is also not a one-in-one-out ordering.

Corollary 5.0.12 *Every LexBFS ordering of a graph G is a one-in-one-out ordering if and only if G is a cograph.*

Chapter 6

Contributions: The Structure of Prime (P_5, $\overline{P}_5$)-free Graphs

The fast recognition algorithms in Chapter 4 depend on a structural characterization of the prime graphs of the considered graph class. Hence, the investigation of properties of prime (P_5, $\overline{P}_5$)-free graphs is worthwhile if one would like to improve the recognition time of these graphs through a modular decomposition.

6.1 Prime Non-Split (P_5, $\overline{P}_5$, C_5)-free Graphs

Recall that a graph is *split* if and only if it is (C_4, C_5, $2K_2$)-free, where a $2K_2$ is the complement of a C_4. Since $2K_2$ and C_4 are subgraphs of P_5 and $\overline{P}_5$, respectively, split graphs are (P_5, $\overline{P}_5$, C_5)-free so we are interested in properties of graphs that are (P_5, $\overline{P}_5$, C_5)-free but not split. A theorem related to P_4-structure (a graph concept which is not discussed here; the interested reader is referred to [31]) is of use to us here:

Theorem 6.1.1 *[31] Let G be a prime graph. Then either every vertex of G is in a P_4 or G is split.*

When considering prime non-split graphs, we are thus able to partition the vertex set into the three disjoint sets based on how a vertex appears in P_4s.

Definition 6.1.2 *Let $G = (V, E)$ be a prime non-split graph. Define $V = V_E \cup V_M \cup V_B$ where*

- *$v \in V_E$ if v is an end of all P_4s it is contained in,*
- *$v \in V_M$ if v is a midpoint of all P_4s it is contained in,*
- *$v \in V_B$ if v is an end of some P_4 and a midpoint of some P_4,*

Note that the three defined sets are disjoint and their union contains all vertices, so they serve as a proper partition of the vertex set.

We can immediately begin to state properties of some of these individual sets by using a theorem of Hoàng and Khouzam. We state it here in terms of our notation:

Theorem 6.1.3 *[34] Let x be a vertex in V_E (respectively, V_M) of a prime graph G. Then x is simplicial in G (simplicial in $\overline{G}$.)*

It is not hard to see that the converse is true. For instance, a simplicial vertex can never be a midpoint of a P_4 and so simplicial vertices must belong to V_E. Using Lemma 2.4.2 and Corollary 2.4.3, we have that in a prime non-split graph, V_E forms the set of simplicial vertices and is a stable set, while V_M is the set of co-simplicial vertices and is a clique. All that can be said about V_B at this point is that it contains those vertices which are not simplicial nor co-simplicial. Note that under taking the complement of a graph, the sets V_E and V_M switch roles while the vertices in V_B remain in the same set.

Hoàng and Khouzam prove, in the same paper, a theorem regarding the existence of simplicial vertices.

Theorem 6.1.4 *[34] Let G be a prime HHD-free graph. Then G contains two nonadjacent simplicial vertices.*

Since (P_5, $\overline{P}_5$, C_5)-free graphs are HHD-free graphs we can apply this theorem to prime (P_5, $\overline{P}_5$, C_5)-free graphs. Noting that the complement of a (P_5, $\overline{P}_5$, C_5)-free graph is an HHD-free thus allows us to assert the following:

Corollary 6.1.5 *Let G be a prime non-split (P_5, $\overline{P}_5$, C_5)-free graph. Then $|V_E| \geq 2$ and $|V_M| \geq 2$.*

This follows since we will not have a vertex which is both simplicial and co-simplicial in a non-split graph. If a vertex was simplicial and co-simplicial, then its neighbourhood is a clique and its non-neighbourhood is a stable set, which would imply the graph is split.

We can establish a similar property on the V_B set by the use of minimal prime extensions. Since we are considering non-split graphs, such a graph must have at least one of a C_4, a $2K_2$ or a C_5, so a non-split (P_5, $\overline{P}_5$, C_5)-free graph must have one of a C_4 or $2K_2$. Theorem 2.3.4 tells us that if a prime graph contains a C_4 then it must contain at least one of an H_6, D_6 or $\overline{P}_5$ but note that (P_5, $\overline{P}_5$, C_5)-free graphs can not have a D_6 nor a $\overline{P}_5$. This means that if there is a C_4 in such a prime graph we must also have an H_6, and if there is a $2K_2$ there must exist an $\overline{H}_6$.

Corollary 6.1.6 *Let G be a prime non-split (P_5, $\overline{P}_5$, C_5)-free graph. Then G contains H_6 or $\overline{H}_6$.*

The H_6 and $\overline{H}_6$ each have two vertices which are both endpoints and midpoints of a P_4 and so these two vertices will be in V_B and thus $|V_B| \geq 2$.

6.2 Relationships Between Vertex Partition Sets

Here, we accumulate information on how the vertices in V_E, V_M, and V_B relate to each other. In particular, the nature of the V_B set is of interest.

We will be reasoning about vertices around induced P_4s often, so the following lemma will be useful to shorten proofs.

Lemma 6.2.1 *Let G be a $(P_5, \overline{P}_5, C_5)$-free graph and the vertex subset $\{a, b, c, d\}$ induces a P_4 in G. If some vertex v sees a and misses b then v must also see c and miss d.*

Proof. First note that v must see at least one of the vertices in the P_4 since otherwise it forms a P_5. If v sees d, this forms a $\overline{P}_5$ (respectively, a C_5) if it also sees (respectively, does not see) vertex c. Since both $\overline{P}_5$ and C_5 are forbidden, v cannot see d, and so must see only c. □

6.2.1 Refinement of the V_B-Vertex Set

When a vertex sees every vertex in a set S, we say that vertex is *universal* on S. If the vertex sees none of the vertices in S, then we say it is *null* on S. If a vertex sees some of S and misses some of S, then we say it is *partial* on S. Say that a vertex v in V_B belongs to the set B_{ij} where $i, j \in \{0, 1, *\}$ and $i = 0, 1$, or $*$ if v is, respectively, null, universal, or partial on V_E, and j is defined similarly depending on the adjacency with V_M. Thus, these are the possible sets a vertex in V_B can belong to, and the associated description of each set:

B_{11} = vertices of V_B universal on V_E and universal on V_M

B_{1*} = vertices of V_B universal on V_E and partial on V_M

B_{10} = vertices of V_B universal on V_E and null on V_M

B_{*1} = vertices of V_B partial on V_E and universal on V_M

B_{**} = vertices of V_B partial on V_E and partial on V_M

B_{*0} = vertices of V_B partial on V_E and null on V_M

B_{01} = vertices of V_B null on V_E and universal on V_M

B_{0*} = vertices of V_B null on V_E and partial on V_M

B_{00} = vertices of V_B null on V_E and null on V_M

We claim that most of these sets are in fact empty and that V_B can be partitioned into exactly three of the above types.

To show this claim, we will only require the knowledge that vertices in V_E (resp. V_M) are simplicial (resp. co-simplicial) for these properties.

Proposition 6.2.2 *Assume some $b \in V_B$ in a prime non-split (P_5, $\overline{P}_5$)-free graph sees some $e \in V_E$. Then b is V_M-universal.*

Proof. Assume, on the contrary, that b misses some $m \in V_M$. Since m is cosimplicial, it cannot miss the $\{b, e\}$ edge, so m must see e. But since e is simplicial, all of its neighbours must be adjacent, and so b must see m. □

In terms of refining the set V_B, we have:

Corollary 6.2.3 *Each of the sets $B_{10}, B_{1*}, B_{*0}, B_{**}$ is empty.*

The above property in the complement graph happens to be equivalent to the contrapositive statement, but we state it as it is still worth noting:

Corollary 6.2.4 *If some $b \in V_B$ misses a vertex in V_M, then b is V_E-null.*

Next, we show that B_{00} is empty:

Proposition 6.2.5 *No vertex $b \in V_B$ in a prime non-split (P_5, $\overline{P}_5$)-free graph is V_E-null and V_M-null.*

Proof. Assume the set B_{00} is nonempty, and let b_1 be a vertex in B_{00} of smallest degree. Since it is not simplicial it must see two nonadjacent vertices, say {u,v}. Since b_1 sees no vertices in $V_E \cup V_M$, this non-edge must also be from the V_B set. Every $m \in V_M$ is cosimplicial and misses b_1, so every m is universal on $N(b_1)$. The pair $\{u, v\}$ must have some vertex distinguishing them since our graph is prime, so say vertex b_2 misses u and sees v. Since b_2 misses u, it is not from the V_M set. Note that if b_2 sees any $m \in V_M$ then we would have a $\overline{P}_5$, so b_2 can not be a vertex from V_E or from $B_{0*} \cup B_{01} \cup B_{11}$. So b_2 is also from the B_{00} set. If b_1 is adjacent to b_2, then there is a $\overline{P}_5$.

Since b_1 was chosen as a vertex from B_{00} with smallest degree, b_2 must see some vertex z that b_1 does not. If z misses v, we have the P_4 u, b_1, v, b_2 with z seeing b_2 and missing v, so by lemma 6.2.1 z must see b_1. So if b_1 misses some $z \in N(b_2)$ this z must see v. Every m is universal on $N(b_2)$ so if z misses u then for any m in V_M, $\{b_1, u, v, z, m\}$ induces a $\overline{P}_5$. We must then have that z sees both u and v. But now $\{b_1, u, v, b_2, z\}$ induces a $\overline{P}_5$, and so it must be the case that b_1 dominates b_2, contradicting the fact that b_1 was chosen with minimal degree. □

The complementary property eliminates yet another subset of V_B:

Corollary 6.2.6 *The set B_{11} is empty.*

Thus the set V_B is reduced to only three types of vertices, corresponding to the sets B_{01}, B_{0*}, B_{*1}. The set of vertices in the first set remain in that set under graph complementation, while the vertices in the other two sets swap with each other.

6.2.2 P_4-wing Orientations

Because our vertices are characterized by how they belong in P_4s, it would be beneficial to investigate how the P_4s exist around the vertices. Recall that the two edges of a P_4 are called the *wings* of the P_4. As a way of encoding information, for each P_4 orient the wings inward (from an endpoint to its adjacent midpoint.) This should not be confused with an oriented graph or a digraph, as there may be some edges which are not oriented and some edges that may be oriented in both directions. With respect to the vertex types introduced, we see that any vertex in V_E can only have incident edges oriented away from it, any vertex in V_M can only have incident edges oriented towards it, and any vertex in V_B must have at least one edge oriented away from it and at least one edge oriented towards it, with no restriction on both these orientations possibly coming from the same edge. For brevity, we will use the notation $a \to b$ to refer to the oriented edge from a to b, and $a \leftrightarrow b$ for a doubly-oriented edge. Having an edge $a \to b$ does not exclude the possibility that the same edge may also be oriented from b to a.

This wing orientation will help us prove a key property regarding vertices in V_E and V_M. Clearly, in any prime graph there must be a P_4. In our case of non-split (P_5, $\overline{P}_5$, C_5)-free prime graphs we can say something stronger.

Theorem 6.2.7 *Let G be a prime (P_5, $\overline{P}_5$, C_5)-free non-split graph. Then there exists a P_4 in G whose endpoints are in V_E and whose midpoints are in V_M.*

Note that this theorem is sharp in that it does not hold for the prime non-split graphs C_5, the P_5 or the $\overline{P}_5$ (the C_5 itself would have all vertices in V_B, the P_5 has only one V_M vertex, and the $\overline{P}_5$ has only one V_E vertex.)

To prove Theorem 6.2.7, first notice that it would suffice to prove that the set of vertices induced by $V_E \cup V_M$ is always connected. To see why this is sufficient, let $S = V_E \cup V_M$, and notice that S is complement-independent. Proving that S is connected also proves that S is co-connected, so S cannot induce a cograph and thus must contain a P_4. To prove that S is connected, recall that it is a split graph with V_M forming a clique. Hence it would be sufficient to prove that every vertex in V_E sees some vertex in V_M. To prove Theorem 6.2.7 we can then prove this stronger theorem instead:

Theorem 6.2.8 *In a prime non-split (P_5, $\overline{P}_5$, C_5)-free graph, every vertex of V_E sees some vertex in V_M.*

Proof. Let $e \in V_E$. Assume on the contrary that e sees no vertex in V_M. Since the V_E set is stable, e must only see vertices in V_B. Let the neighbourhood of e, $N(e)$, be partitioned into N_1 and N_2 where N_1 are the vertices $b_i \in N(e)$ for which the edge $\{e, b_i\}$ is oriented $e \to b_i$, and the set N_2 is the set of vertices for which the edges joining e and any vertex in N_2 are not oriented.

Since e is simplicial, $N(e)$ is a clique as is N_1 in particular. Since $N_1 \subset V_B$, every vertex in N_1 must have an out-oriented edge. We now require some lemmas to proceed. In an H_6, let the edge joining the two vertices of degree two be called the *bottom edge*, or simply the *bottom*, of the H_6.

Lemma 6.2.9 *If $x \leftrightarrow y$ then $\{x, y\}$ is the bottom of some H_6.*

Proof. Since $x \to y$, there must be vertices a and b such that (x, y, a, b) (as an ordered set) forms a P_4, and since $y \to x$ we also have vertices c and d so that (y, x, c, d) is a P_4. Notice that either a, b, c and d are either all distinct or we have b coinciding with d. If, indeed, $b = d$ then the P_4s (x, y, a, b) and (y, x, c, b) would form a C_5 unless c sees a in which case we have a $\overline{P}_5$. Thus it must be the case that b and d are distinct.

Since (x, y, a, b) forms a P_4 and vertex c sees x and not y, lemma 6.2.1 applies so c must see a and miss b. Notice that if d sees a then we have a $\overline{P}_5$ and if d sees b then we have a P_5. Hence edge $\{x, y\}$ is the bottom of an H_6. □

This lemma is useful to us here since every vertex in N_1 must have an out-edge and now we can assert that there are no doubly-oriented edges in N_1. If there is an edge $\{x, y\} \in N_1$ which is doubly-oriented, lemma 6.2.9 tells us it is the bottom of an H_6. Since e and everything in $N(e)$ see both x and y, the $\{x, y\}$ edge must form the bottom of an H_6 with four vertices outside of $N(e) \cup \{e\}$. Since $\{x, y\}$ forms a C_4 with two vertices outside of $N(e)$, and e sees both x and y, we have a house, which is not allowed.

We need a second lemma on this orientation to continue:

Lemma 6.2.10 *Assume vertex $a \to b$ and $b \to c$. If there is no $\{a, c\}$ edge, then $a \leftrightarrow b$. (See Figure 6.1)*

Proof. Since $b \to c$, there must be a P_4 (b, c, x, y), and note that both x and y must be distinct from a. Since a sees b and misses c, we apply 6.2.1 and so a sees x and misses y. Since b must miss both x and y, we have the P_4 (b, a, x, y) and so $b \to a$. □

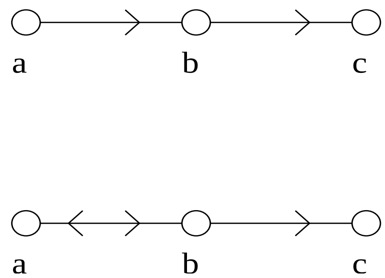

Figure 6.1: Orientations corresponding to Lemma 6.2.10

Since every vertex in N_1 has an out-edge, we now know every such out-edge cannot end at a vertex outside $N(e)$, as the above lemma tells us that e is on a doubly-oriented edge which is impossible for a vertex in V_E. Hence every oriented edge from N_1 leads to a vertex in $N(e)$. We will prove something stronger, that every oriented edge from N_1 leads to a vertex in N_1. The following lemma asserts that if there was an edge oriented from N_1 to a vertex v in N_2 then it must be that $e \to v$, but this contradicts $v \in N_2$.

Lemma 6.2.11 *Let $a \to b$ and $b \to c$. If there is an $\{a, c\}$ edge, then either $a \leftrightarrow b$ or $a \to c$.*

Proof. Again, we have the vertices x and y so that (b, c, x, y), with x and y distinct from a. We assume that $\{a, b\}$ is not doubly oriented and prove that $a \to c$. If a misses both x and y then we are done, and if a sees y only, we have a $\overline{P}_5$ so it must be the case that a sees x. Now a must also see y since if it did not, the $\{a, b\}$ edge would be doubly oriented due to the P_4 (b, a, x, y).

Since $a \to b$, we need u and v such that (a, b, u, v) forms a P_4. Now we have a seeing all of $\{b, c, x, y\}$ so we know that u and v must be distinct from x and y. To avoid a P_5, u will have to see some of $\{c, x, y\}$. Note that if u sees either x or y, then it must see both x and y or else we have a $\overline{P}_5$ or a C_5. Since (a, b, u, v) is a P_4 and y sees a, then Lemma 6.2.1 tells us that y sees u and misses v. So u sees both x and y and now must also see c or else a $\overline{P}_5$ is formed.

If we now prove v misses c, then the P_4 (a, c, u, v) provides the $a \to c$ property sought for. If it was the case that v sees c, v would have to also see y or else we would have the $\overline{P}_5$ $\{a, c, u, v, y\}$. But now there is the forced $\overline{P}_5$ on the vertices $\{a, b, u, v, y\}$, and so v cannot see c and thus $a \to c$. □

Corollary 6.2.12 *If $a \to b$, $b \to c$ and $c \to a$, then there must be either $a \leftrightarrow b$ or $a \leftrightarrow c$.*

When looking at the neighbourhood $N_1 \cup N_2$ of a simplicial vertex, Lemma 6.2.11 tells us that there will not be oriented edges from N_1 to N_2. Combining this with the previous lemmas tells us that all the oriented edges in N_1 must terminate in N_1 and so the oriented vertices in N_1 must have a directed cycle.

Corollary 6.2.12 follows from Lemma 6.2.11 just by adding the extra oriented edge $c \to a$ in the hypothesis, and using Lemma 6.2.9 we can see that N_1 can not have any oriented triangles. Assume there are oriented cycles in N_1 on more than three vertices, so $a \to b$ and $b \to c$ without c pointing to a. Recall that N_1 is a clique, and so there must be an $\{a, c\}$ edge. Lemma 6.2.11 tells us that either we have a doubly-oriented edge, which is not allowed, or $a \to c$. This shows that any oriented cycle on n vertices implies the existence of an oriented cycle on $n-1$ vertices, and in particular an oriented triangle which has already been shown to not exist [1].

We have thus shown that N_1 can not accommodate the arrows that must exist in the case that $N(e) \subseteq V_B$, and so any $e \in V_E$ must see some vertex of V_M, as required. □

Due to prior discussion, the proof of Theorem 6.2.8 also establishes Theorem 6.2.7, that a non-split prime (P_5, $\overline{P}_5$, C_5)-free graph has a P_4 whose endpoints are in V_E and whose midpoints are in V_M.

6.3 Structures Within Prime (P_5, $\overline{P}_5$)-free Graphs

Recall that any prime graph must contain a P_4, and we just showed that a prime (P_5, $\overline{P}_5$)-free graph must contain a P_4 of a particular type - one whose vertices belong to the V_M and V_E sets. Since we are considering non-split graphs, we know there is some H_6 or $\overline{H}_6$ in our graph as well. We believe that prime (P_5, $\overline{P}_5$)-free graphs contain a special type of H_6 or $\overline{H}_6$ as well:

Conjecture 6.3.1 *Let G be a non-split prime (P_5, $\overline{P}_5$)-free graph. Then in G or $\overline{G}$ there exists an H_6 whose two vertices of degree one (the top vertices) are end-P_4 only and whose neighbouring vertices are mid-P_4 only.*

Much of the investigation into the V_E, V_M, and V_B sets was motivated by this conjecture. The lack of knowledge in characterising the V_B set is an obstacle in the way of proving Conjecture 6.3.1. For example, if we let $|V_E| = |V_M| = 2$ the conjecture still resists proof, even with the knowledge that the four vertices in $V_E \cup V_M$ must induce a P_4. We hope that this difficulty can add to the appreciation of the refinement properties of the V_B.

One observation that can be made regarding V_E-V_M-type P_4s extending to H_6s and $\overline{H}_6$s is that such a P_4 is exclusive to one type. Namely,

[1] One can show that the P_4-wing orientation is in fact acyclic in general, but the proof is longer and not required for our needs here, as we only need that the N_1 portion of the neighbourhood of a simplicial vertex has no doubly-oriented edges.

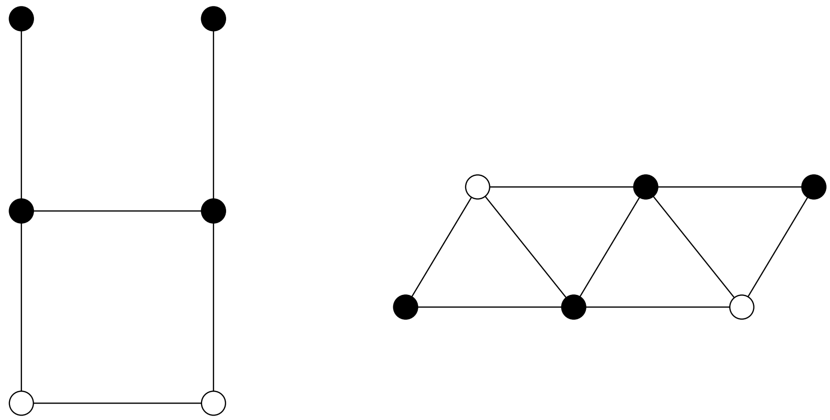

Figure 6.2: The arms of an H_6 and the tunnel of an $\overline{H}_6$ (induced by solid vertices)

Theorem 6.3.2 *Let G be a non-split prime (P_5, $\overline{P}_5$)-free graph and E_1, M_1, M_2, E_2 induce a P_4 with $E_1, E_2 \in V_E$ and $M_1, M_2 \in V_M$. Then there can not be vertices u, v, x, y such that $\{E_1, M_1, M_2, E_2, u, v\}$ and $\{E_1, M_1, M_2, E_2, x, y\}$ induce an H_6 and $\overline{H}_6$ simultaneously.*

Just as we showed that every vertex in V_E sees some vertex in V_M by proving a stronger theorem, we shall show Theorem 6.3.2 similarly. First, we introduce some terminology. There are several P_4s in an H_6 and so we want to give a name to a specific P_4 to simplify discussion.

Definition 6.3.3 *The* arms *of an H_6 is the unique P_4 in the H_6 whose endpoints are the two vertices of degree one. The* tunnel *of an $\overline{H}_6$ is the complement of an arms. (See Figure 6.2)*

Now we prove a strengthening of Theorem 6.3.2.

Lemma 6.3.4 *Let G be a prime non-split (P_5, $\overline{P}_5$, C_5)-free graph, and let $\{a, b, c, d\}$ induce a P_4 in G. Then $\{a, b, c, d\}$ cannot simultaneously be the arms of an H_6 and the tunnel of an $\overline{H}_6$.*

Proof. Assume some u and v exist such that they form an H_6 with the four vertices in the P_4 with the edges $\{u, b\}$ and $\{v, c\}$. Assume x, y extend the P_4 to an $\overline{H}_6$, with x adjacent to a, b, c and y adjacent to b, c, d. Note that x, y must be distinct from u, v since u and v must each miss one of the mid-P_4 vertices, while x and y do not. Then u, v, b, c, x induce a $\overline{P}_5$, unless x sees at least one of u or v. If x sees v, then we must have that y sees v as well or else d, y, b, x, v will form a P_5. But then a, x, v, y, b must induce a $\overline{P}_5$, and so it cannot be the case that x sees v. Similarly, y cannot see u and so x sees u and similarly, y sees v. But then a, x, u, v, y is a P_5, and this is unavoidable. Hence it is impossible that such u, v, x, y exist. □

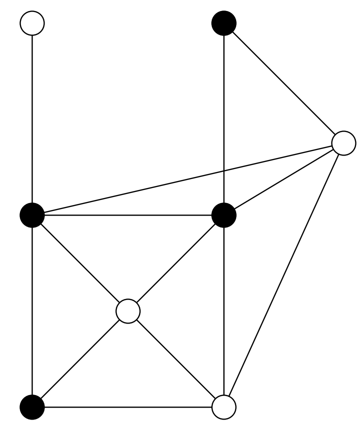

Figure 6.3: A prime (P_5, $\overline{P}_5$)-free graph with a P_4 in both an H_6 and an $\overline{H}_6$

This proves the theorem as well, when combined with the observation that the only way a P_4 from $V_E \cup V_M$ can exist in an H_6 or $\overline{H}_6$ is by being the arms of the H_6 or the tunnel of the $\overline{H}_6$.

Lemma 6.3.4 can not be strengthened by removing the arms or tunnel restriction on the P_4. Figure 6.3 shows a prime (P_5, $\overline{P}_5$)-free graph with a P_4 extending to an H_6 as well as an $\overline{H}_6$.

6.4 Towards the H_6-Conjecture

The H_6-conjecture (Conjecture 6.3.1) was the primary focus during much of this research, and from it sprung a host of properties regarding vertices in prime non-split (P_5, $\overline{P}_5$, C_5)-free graphs. This chapter presents the key results.

6.4.1 Adjacency Properties

One of the theorems already proven was that every vertex in V_E sees some vertex in V_M (Theorem 6.2.8.) The complementary property gives another adjacency rule. Recall that the vertices in V_E are the simplicial vertices and the V_M vertices are the cosimplicial vertices.

Corollary 6.4.1 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, every vertex in* V_M *misses some vertex in* V_E*.*

Also recall that every vertex in V_B falls into exactly one of the sets B_{*1}, B_{01} or B_{0*}. This translates to: a vertex in V_B is either partial on V_E and universal on V_M, null on V_E and universal on V_M, or null on V_E and partial on V_M, respectively. Hence we have already proven the following property:

Corollary 6.4.2 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, every vertex in* V_B *sees a vertex in* V_M *and misses a vertex in* V_E*.*

More vertex-adjacency properties follow:

Proposition 6.4.3 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, every vertex in* V_M *sees a vertex in* V_B*.*

Proof. Assume some m in V_M is V_B-null. This implies B_{*1} and B_{01} are empty, and so any b vertex must be of type B_{0*}. Let b be a vertex in V_B. Since b is the mid-vertex of some P_4 it must be adjacent to a vertex, v, which is the end of that P_4. Now v can not be in V_E since b is from B_{0*}, v can not be from V_M since those vertices are never the end of a P_4, thus v must be another vertex from V_B. But then $\{b, v\}$ forms an edge that m misses, contradicting the fact that it is cosimplicial. □

Corollary 6.4.4 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, every vertex in* V_E *misses a vertex in* V_B*.*

It was observed earlier that if a P_4 from $V_E \cup V_M$ was in an H_6, then it must be the arms of the H_6. Further note that in order for this P_4 to extend to an H_6, it would be necessary for there to be two adjacent vertices from B_{0*}. As a possible step towards proving the H_6 conjecture, a B_{0*} edge is a good starting point.

Proposition 6.4.5 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, every edge* $\{x, y\}$ *in* B_{0*} *extends to a* C_4 *with two vertices from* V_M*.*

Proof. Every vertex in B_{0*} sees a vertex and misses a vertex in V_M. Say x misses $m_1 \in V_M$ and y misses $m_2 \in V_M$. Since m_1 is cosimplicial, it can not miss the $\{x, y\}$ edge, so m_1 sees y and similarly, m_2 sees x. Since V_M forms a clique, m_1 sees m_2 and this forms a C_4. □

Corollary 6.4.6 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, every pair of nonadjacent vertices in* B_{*1} *extends to a* $2K_2$ *with two vertices from* V_E*.*

This proposition has not helped resolve the H_6 conjecture, but it has lead to a other interesting properties.

Proposition 6.4.7 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, the set* B_{0*} *is* $2K_2$*-free.*

Proof. Let $\{x, y\}$ and $\{v, w\}$ be the two edges of the $2K_2$. Using 6.4.5, we have m_1 adjacent to y and m_2 adjacent to x, forming a C_4. Since the cosimplicial vertices cannot miss the $\{v, w\}$ edge, let m_1 see v. Also, m_2 must see v or w, and if m_2 sees v then $\{x, y, m_1, m_2, v\}$ forms a $\overline{P}_5$. So m_2 must miss v and see w. If m_1 sees w then $\{x, y, m_1, m_2, w\}$ is a $\overline{P}_5$, but if instead m_1 misses w then w, v, m_1, y, x is a P_5. □

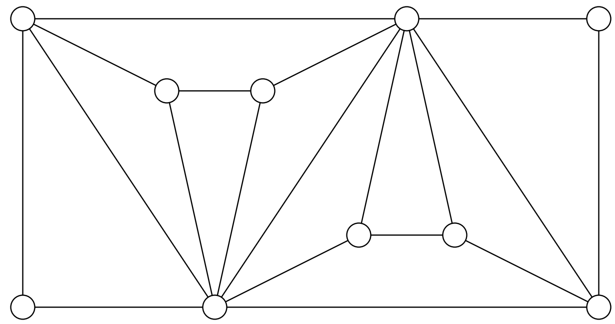

Figure 6.4: A prime (P_5, $\overline{P}_5$)-free graph G with no vertex of degree one in G or $\overline{G}$

Corollary 6.4.8 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, the set* B_{*1} *is* C_4*-free.*

Proposition 6.4.9 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, every vertex of* B_{01} *sees at least one vertex of every edge in* B_{0*}*.*

Proof. Proposition 6.4.5 gives us a C_4 on any B_{0*} edge, and since a b from B_{01} sees both V_M vertices of the C_4, we will have a $\overline{P}_5$ unless b sees some of the B_{0*} edge. □

Corollary 6.4.10 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, every vertex of* B_{01} *misses at least one vertex of every non-edge pair in* B_{*1}*.*

Proposition 6.4.11 *In a prime non-split* (P_5, $\overline{P}_5$, C_5)*-free graph, either* B_{0*} *is a stable set or* B_{*1} *is a clique.*

Proof. Suppose neither of the two holds, so B_{0*} has an edge $\{v_1, v_2\}$ and B_{*1} has a non-edge $\{w_1, w_2\}$. Proposition 6.4.5 tells us that there are m_1 and m_2 from V_M such that v_1, v_2, m_2, m_1 is a C_4, and similarly we have e_1 and e_2 from V_E forming a $2K_2$ with edges $\{w_1, e_1\}, \{w_2, e_2\}$. Since w_1 and w_2 see both m_1 and m_2, there is a P_5 (e_1, w_1, m_i, w_2, e_2), for $i = 1, 2$ unless m_1 and m_2 are seen by e_1 and/or e_2. Now, there can not be any vertex from V_E seeing both m_1 and m_2 or else we have a $\overline{P}_5$, so e_1 and e_2 must form a P_4 with m_1 and m_2. Notice then that $\{v_1, v_2, m_1, m_2, e_1, e_2\}$ is an H_6 with $\{e_1, e_2, m_1, m_2\}$ forming its arms, while this P_4 is also the tunnel of the $\overline{H}_6$ formed by $\{w_1, w_2, e_1, e_2, m_1, m_2\}$. Lemma 6.3.4 tells us that this is impossible. □

On small (nine or less vertices) prime non-split, (P_5, $\overline{P}_5$, C_5)-free graphs, every graph G has the property that either G or $\overline{G}$ has a vertex of degree one. Note that a vertex of degree one is always simplicial, and so its single neighbour must be a vertex from V_M. A vertex of degree one will never be in a C_5 or $\overline{P}_5$, and the only way it can be in a P_5 is if it is an endpoint of the P_5. The neighbour of an endpoint of a P_5 is an end-P_4 vertex, but

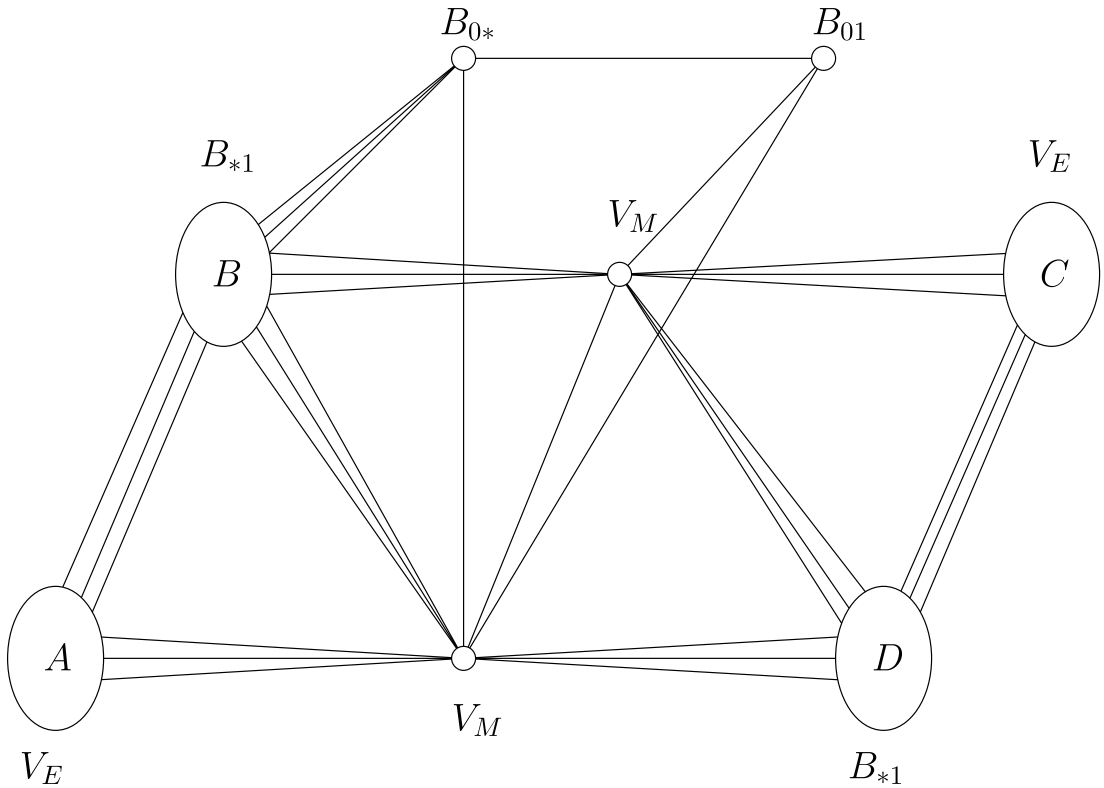

Figure 6.5: A prime (P_5, $\overline{P}_5$, C_5)-free graph with a vertex of smallest degree that is not simplicial

the only neighbour of the degree one vertex in our graph is a vertex from V_M, implying that the degree one vertex can not be the end of a P_5. Hence, degree one vertices (in the graph or in its complement) can always be removed from consideration when testing for (P_5, $\overline{P}_5$, C_5)-free graphs. It would be nice if it were true that there is always such a vertex, but this fails when looking at graphs on ten vertices. That is, the smallest prime non-split (P_5, $\overline{P}_5$, C_5)-free graphs G with no degree one vertex in G or $\overline{G}$ have ten vertices.

Counterexample 6.4.12 *Figure 6.4 is a prime non-split* (P_5, $\overline{P}_5$)*-free graph G without a degree one vertex in G or $\overline{G}$.*

In the counterexample graph, the vertices of smallest degree are still simplicial. One might wonder if the vertices of minimum degree are always simplicial. Figures 6.5 and 6.6 depict a family of graphs on at least 16 vertices with minimum degree three, one such vertex being from the V_B set. In Figure 6.5, the sets A, B, C, D represent collections of vertices, A and C being stable sets while B and D being cliques. $|A| = |B|$ and must have at least three vertices each. A *perfect matching* between two sets of vertices is a set of edges forming a one-to-one correspondence between the two sets. The vertices of A and B are joined by the complement of a perfect matching, as is shown in Figure 6.6. The adjacency of a vertex in figure 6.5 to one of the sets A, B, C, D represents a complete adjacency from that vertex to

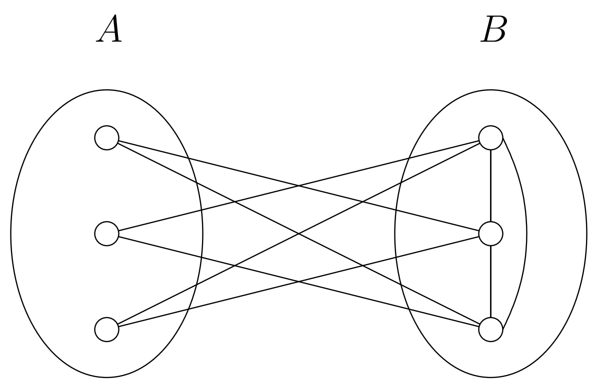

Figure 6.6: The complement of a perfect matching joining sets A and B from Figure 6.5

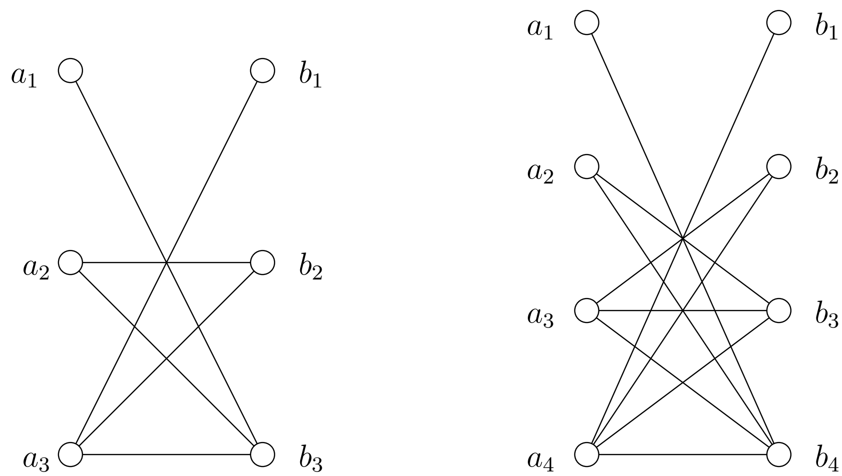

Figure 6.7: Some prime P_5-free bipartite graphs

every vertex of the set. When each of the four sets contain exactly three vertices each, there are many vertices of degree three in the graph, including the V_B vertex of type B_{01}. To have the B_{01} as the only vertex with smallest degree, we can increase the sizes of A, B, C, D to four each, giving the resulting graph 20 vertices in total. This is the smallest known example showing that a vertex of minimum degree is not simplicial.

6.4.2 Bipartite Substructures

Recalling the theorems of Fouquet (Theorem 4.1.2) and Fouquet and Giakoumakis (Theorem 4.2.3,) P_5-free bipartite graphs play an important role in recognizing classes of (P_5, $\overline{P}_5$)-free graphs. The prime graphs have a simple description and can be recognized in linear time [19]. In each (stable set) partition of a prime P_5-free bipartite graph, there is always exactly one vertex of degree i for all i from 1 to $|V|/2$ [19]. Examples of prime P_5-free bipartite graphs are given in Figure 6.7.

We will define some notation to make our discussion of these graphs simpler. Let the bipartitions of a prime P_5-free bipartite graph be the stable sets A and B, and the vertices in these respective sets will be called a_i and b_j, where the subscripts coincide with the degree of the vertex. So, for example, vertex a_3 sees exactly three vertices from B. We will call a prime P_5-free bipartite graph with $2n$ vertices a $P_{n,n}$. Note that P_4 is equal to $P_{2,2}$ and H_6 is $P_{3,3}$.

It is interesting to see how vertices in a prime (P_5, $\overline{P}_5$, C_5)-free graphs form around one of these $P_{n,n}$ subgraphs.

Lemma 6.4.13 *Let H be a subgraph of a prime (P_5, $\overline{P}_5$, C_5)-free graph isomorphic to a $P_{n,n}$ for some $n \geq 3$, and let v be a vertex not in H that sees some a_k of H and misses all vertices in B. Then v sees a_i for all i from $k+1$ to n.*

Proof. Note that a_k must see the k vertices in B of highest degree, so if a_k misses any vertices in B it must miss $b_1, b_2, \ldots, b_{n-k}$. Assume v misses some a_t for some $k+1 < t \leq n$. Now v, a_k, b_n, a_t, b_1 form a P_5, so v must see a_t. □

If two vertices have the same neighbourhood, they are called *twins*. Twins may or may not be adjacent. The above lemma showed that v is the nonadjacent twin of b_j for some j from $n-k+1$ to n.

Lemma 6.4.14 *Let H be a subgraph of a prime (P_5, $\overline{P}_5$, C_5)-free graph isomorphic to a $P_{n,n}$ for some $n \geq 3$, and let T be the set of all twins of a_1 with respect to H. Then T forms a stable set.*

Proof. Assume T has an edge. There must be a distinguisher which is partial on some edge of T, and this distinguisher can not be in H. Let d see t_1 and miss t_2 where $\{t_1, t_2\}$ is an edge in T. Since d is not in T, it must miss b_n. Now t_1, b_n, a_n, b_j induces a P_4 for every $1 \geq j \geq n-1$, and since d sees t_1 and misses b_n, lemma 6.2.1 tells us that d must see miss every b_j for $1 \geq j \geq n-1$. Also, t_1, b_n, a_i, b_{n-1} forms a P_4 for $2 \geq i \geq n$, so the lemma says that d sees all such a_i. Now t_2, t_1, d, b_n, a_2 is a $\overline{P}_5$, a contradiction. □

We end this chapter with an example showing that identifying maximal $P_{n,n}$ structures is not sufficient to find two V_E vertices in an H_6.

Counterexample 6.4.15 *If H is a maximal $P_{n,n}$ in a prime non-split (P_5, $\overline{P}_5$, C_5)-free graph G, then the vertices a_1 and b_1 are not necessarily simplicial in G, as shown in Figure 6.8.*

What can be shown, however, is that the neighbourhood of a_1 will not have a non-edge containing b_n.

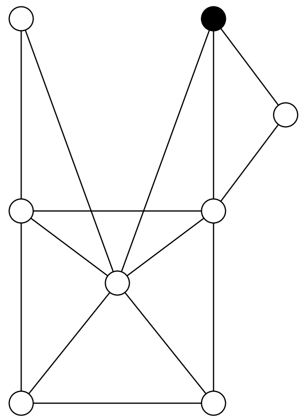

Figure 6.8: In a maximal $P_{3,3}$, an a_1 that is not simplicial

Proposition 6.4.16 *If H is a maximal $P_{n,n}$ in a prime non-split (P_5, $\overline{P}_5$, C_5)-free graph and v is a vertex not in H but sees a_1 in H, then it must also see b_n.*

Proof. Assume v misses b_n. The proof of the previous lemma used 6.2.1 to show that v must see all a_i and miss all b_j, so v is a twin of b_n. Some distinguisher d must be partial on $\{v, b_n\}$, but then $H \cup \{v, d\}$ induces a $P_{n+1,n+1}$. □

Interpreting a P_4 as a $P_{2,2}$ and generalizing this to larger $P_{n,n}$ forms all bipartite graphs in our class. Consider the operation that takes any V_M vertex m, creates a nonadjacent twin of m, and then adds a new vertex of degree one distinguishing m and its twin. Given a prime non-split (P_5, $\overline{P}_5$, C_5)-free graph G, call G^m the resultant graph after applying this operation on G and m.

Proposition 6.4.17 *Let G be a prime non-split (P_5, $\overline{P}_5$, C_5)-free graph and m be a vertex in V_M. Then G^m is prime (P_5, $\overline{P}_5$, C_5)-free .*

Proof. The creation of a twin will not create a P_5, a $\overline{P}_5$ nor a C_5 as those are prime graphs. Since m was cosimplicial, its twin will also see every edge and be cosimplicial as well so it can not be the end of any P_4. The addition of a vertex with degree one could only possibly form a P_5, specifically by being the end of a P_5. But since its only neighbour is not the end of a P_4, we can be sure that no P_5 is created, so G^m is (P_5, $\overline{P}_5$, C_5)-free . □

When applying this operation to $P_{n,n}$ it creates $P_{n+1,n+1}$. This operation need not be applied to just the non-split graphs; for instance, it can be applied to the bull and still create a prime (P_5, $\overline{P}_5$, C_5)-free graph, even when picking the vertex of the bull which is not in a P_4. A P_5 will never be created as long as the vertex chosen to duplicate is not the end of a P_4. If m_1 is the chosen vertex to duplicate, m_2 is its twin, and distinguisher d sees

m_2 and misses m_1, then m_1 becomes the end of a P_4 and so it moves to the V_B set. The distinguisher d belongs to the V_E set, and m_2 is in the V_M set. Any vertices e_1 from V_E adjacent to m_1 before the duplication will see the nonedge $\{m_1, m_2\}$ so they are no longer simplicial and are the midpoint of the P_4 (m_1, e_1, m_2, d). Such vertices, e_1, move to the V_B set, specifically into the B_{0*} set. The complementary operation also applies, which would be to copy a vertex in V_E with an adjacent twin and create a distinguisher which is universal on the graph but misses the new copied vertex.

Clearly, this operation can not generate all prime (P_5, $\overline{P}_5$, C_5)-free graphs since it will always create a graph with a vertex of degree one (or degree $|V| - 2$, in the case of the complementary operation.) Figure 6.4 shows an example of a ten vertex graph without a vertex of degree one or eight.

Chapter 7

Conclusions and Future Research

The recognition problem for perfectly orderable graphs is an NP-complete problem [47] yet their optimization problems are simple to solve when given a perfect order, as described in Chapter 2. (P_5, $\overline{P}_5$)-free and (P_5, $\overline{P}_5$, C_5)-free graph classes form important subclasses of perfectly orderable graphs, having key connections to many other well-studied graph classes discussed in Chapter 3. The techniques used in Chapter 4 to recognize the (P_5, $\overline{P}_5$, bull)-free and semi-P_4-sparse graphs suggested that understanding the structure of prime (P_5, $\overline{P}_5$, C_5)-free graphs could help in recognizing them.

Partitioning the vertex set of prime non-split (P_5, $\overline{P}_5$, C_5)-free graphs into the V_M, V_E and V_B sets is straightforward and can apply to any non-split prime graphs. The V_M and V_E sets are easy to describe, as they form a clique and stable set, respectively. The V_B set was much harder to describe until it was refined down to three possible types of adjacencies with the V_E and V_M sets. Namely, a vertex in V_B either sees a V_E and is universal on V_M or else it misses V_E and can be either partial or universal on V_M. Some adjacency properties within and between the sets were given, for instance B_{0*} is $2K_2$-free and every vertex in B_{01} sees every edge of B_{0*}.

Future refinements could include the property of $V_E \cup V_M$-type P_4-inclusion. For instance, every V_E is the end of some P_4, but there are V_E vertices that are only in P_4s containing vertices from V_B. A full characterization of the nature of V_B would be an asset in a description of prime (P_5, $\overline{P}_5$, C_5)-free graphs. We are inclined to formally state this special case of the H_6 conjecture as an open problem:

Open problem 7.0.18 *Let G be a prime non-split (P_5, $\overline{P}_5$)-free graph with $|V_E| = 2 = |V_M|$. Prove or disprove that there exists an H_6 with arms from $V_E \cup V_M$ in G or $\overline{G}$.*

The P_4-wing orientation lemmas provide a framework for solving problems that may otherwise not fall easily to proofs relying on adjacency properties. The orientation has the power to hold and use a lot of information on just one or two edges, for instance, Lemma 6.2.9 deduces information over six vertices from only one doubly-oriented edge. The orientation lemmas may be instrumental in a characterization of prime (P_5, $\overline{P}_5$, C_5)-free graphs.

Chapter 5 introduced four self-complementary classes of perfectly orderable graphs and gave results only for the PO4 class. The polynomial-time status of the recognition problem is open for all four classes. Some forbidden subgraphs were given for PO4, but is not exhaustive. A forbidden induced subgraph characterization for any of those classes would gain significant attention. The brittle graphs are a subset of PO2 but its precise position in the hierarchy of the four nested classes is unclear. It would be interesting to know this as well.

Finally, we leave open the problem of recognizing (P_5, $\overline{P}_5$)-free and (P_5, $\overline{P}_5$, C_5)-free graphs in o(n^3)-time. Even assuming the H_6 conjecture to be true, there is no clear way of using it towards recognizing the graph class. Furthermore, the (P_5, $\overline{P}_5$, C_5)-free graphs are perfectly orderable, and the vertex degrees are all that is needed to create a Welsh-Powell ordering and thus a perfect ordering of the vertices. Such an ordering is a strong property to have, but aside from finding max clique, max independent set, min clique cover and min colouring, we have found no added benefits from the ordering. It would be interesting to derive some structural properties of, or prove some propositions on, (P_5, $\overline{P}_5$, C_5)-free graphs from a perfect order.

Bibliography

[1] A. Aho, J. Hopcroft and J. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Reading, Massachusettes, 1974.

[2] A. Brandstädt, C. T. Hoàng and J.M. Vanherpe, On minimal prime extensions of a four-vertex graph in a prime graph, Discrete Mathematics 288 (2004), 9-17.

[3] A. Brandstädt, V.B. Le and J.P. Spinrad, *Graph Classes: A Survey*, SIAM Monographs on Discrete Math and Applications, 1999.

[4] M. Burlet and J. Fonlupt, Polynomial algorithm to recognize a Meyniel graph, Progress in combinatorial optimization (Waterloo, Ont., 1982), 69-99, Academic Press, Toronto, Ont., 1984.

[5] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, The Strong Perfect Graph Theorem, Annals of Mathematics, *to appear.*

[6] V. Chvátal, Perfectly ordered graphs, in *Topics on Perfect Graphs* (C. Berge, V. Chvátal, eds.), Ann. Discrete Math., 21 (1984), 279-280.

[7] V. Chvátal, On the P_4-structure of perfect graphs III. Partner Decompositions, J. Comb. Theory (B) 43 1987 349-353.

[8] V. Chvátal, *Linear Programming*, W.H.Freeman, New York, 1983.

[9] V. Chvátal , C. T. Hoàng , N.V.R. Mahadev and D. de Werra, Some classes of perfectly orderable graphs, J. of Graph Theory, 11 (1987), 481-495.

[10] T. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, *Introduction to Algorithms*, The MIT Electrical Engineering and Computer Science Series. The MIT Press, 2nd edition, 2001.

[11] D.G. Corneil, Y. Perl and L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput., 14 (1985), 926-934.

[12] A. Cournier and M. Habib, A new linear algorithm for modular decomposition, Lecture Notes in Comput. Sci., 787 (1994) 68-84.

[13] E. Dahlhaus, P.L. Hammer, F. Maffray and S. Olariu, On domination elimination orderings and domination graphs, Lecture Notes in Comput. Sci., 903 (1994), 81-92.

[14] E. Eschen, R.B. Hayward, J.P. Spinrad and R. Sritharan, Weakly triangulated comparability graphs, SIAM J. Comput. 29 (1999), 378-386

[15] C.M.H. de Figueiredo and K. Vuskovic, Recognition of quasi-Meyniel graphs, Discrete Appl. Math., 113 (2001), 255-260.

[16] S. Földes and P.L. Hammer, Split graphs, Congres. Numer., 19 (1977), 311-315.

[17] J.L. Fouquet, A decomposition for a class of $(P_5, \overline{P}_5)$-free graphs, Discrete Math., 121 (1993), 75-83.

[18] J.L. Fouquet, Re: $(P_5, \overline{P}_5,$ bull)-free graphs, private communication, Jan. 20 (2003)

[19] J.L. Fouquet and V. Giakoumakis, On semi-P_4-sparse graphs, Discrete Math., 165/166 (1997), 277-300.

[20] J.L. Fouquet and J.M. Vanherpe, On $(P_5, \overline{P}_5)$-sparse graphs and other families, Proceedings of CTW04 Workshop on Graphs and Combinatorial Optimization, (2004), 142-146.

[21] M.U. Gerber and V.V. Lozin, On the stable set problem in special P_5-free graphs, Rutcor Research Report, 24 (2000).

[22] V. Giakoumakis, P_4-laden graphs: a new class of brittle graphs, Inform. Process. Lett., 60 (1996), 29-36.

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

[24] M.C. Golumbic and A.N. Trenk, *Tolerance Graphs*, Cambridge Studies in Advanced Mathematics (No. 89), 2004.

[25] M. Grötchel, L. Lovász and A. Schrijver, Polynomial algorithms for perfect graphs. Ann. Discrete Math., 21 (1984), 325-356.

[26] U.I. Gupta, D.T. Lee, and J.Y.T. Leung, Efficient algorithms for interval graphs and circular-arc graphs, Networks, 12 (1982), 459-467.

[27] P.L. Hammer and B. Simeone, The splittance of a graph, Combinatorica, 1 (1981), 275-284.

[28] R.B. Hayward, Weakly triangulated graphs, J. Combin. Theory B, 39 (1985), 200-208.

[29] R.B. Hayward, Murky graphs, J. Combin. Theory B, 49 (1990), 200-235.

[30] R.B. Hayward, C. T. Hoàng , F. Maffray, Optimizing weakly triangulated graphs, Graphs Combin., 5 (1989), 339-349; Erratum, Graphs Combin., 6 (1990), 33-35.

[31] R.B. Hayward, S. Hougardy and B.A. Reed, Polynomial time recognition of P_4-structure, Proc. 13th SODA (2002) 382-389

[32] R.B. Hayward and W. Lenhart, Bichromatic P_4-composition schemes for perfect orderability, Disc. Applied Math., 141 (2003), 161-183.

[33] C.T. Hoàng, Perfect Graphs, Ph.D. thesis, School of Computer Science, McGill University, Montreal (1985).

[34] C.T. Hoàng and N. Khouzam, On brittle graphs, J. Graph Theory, 12 (1988), 391-404.

[35] C.T. Hoàng and B.A. Reed, Some classes of perfectly orderable graphs, J. Graph Theory, 13 (1989), 445-463.

[36] C.T. Hoàng and R. Sritharan, Finding houses and holes in graphs, Theoret. Comput. Sci., 259 (2001), 233-244.

[37] W.L. Hsu and T.H. Ma, Fast and simple algorithms for recognizing chordal comparability graphs and interval graphs, SIAM Journal on Computing, 28 (1999), 1004-1020.

[38] B. Jamison and S. Olariu, On the semi-perfect elimination, Adv. Appl. Math., 9 (1988), 364-376.

[39] B. Jamison and S. Olariu, P_4-reducible graphs - a class of uniquely tree representable graphs, Studies in Appl. Math., 81 (1989), 79-87.

[40] B. Jamison and S. Olariu, On a unique tree representation for P_4-extendible graphs, Discrete Appl. Math., 34 (1991), 151-164.

[41] B. Jamison and S. Olariu, A new class of brittle graphs, Studies in Appl. Math., 81 (1989), 89-92.

[42] L. Khachiyan, A polynomial algorithm in linear programming, Soviet Mathematics Doklady, 20 (1979), 191-194.

[43] G. Lancia, V. Afna, S. Istrail, R. Lippert and R. Schwartz, SNPs Problems, Complexity and Algorithms, ESA 2002, LNCS 2161, Springer-Verlag Berlin Heidelberg (2001), 182-193.

[44] R.M. McConnell and J.P. Spinrad, Linear-time modular decomposition and efficient transitive orientation of comparability graphs, Symposium on Discrete Algorithms, Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms, (1994), 536-545.

[45] F.R. McMorris and D.R. Shier, Representing chordal graphs on $K_{1,n}$, Comment. Math. Univ. Carolin., 24 (1983), 489-494.

[46] H. Meyniel, The graphs whose odd cycles have at least two chords, In: Topics on Perfect Graphs, C. Berge and V. Chvátal eds., Annal of Disc. Math., 21 (1984), 115-119

[47] M. Middendorf and F. Pfeiffer, On the complexity of recognizing perfectly orderable graphs, Discrete Math., 80 (1990), 327-333.

[48] S. Olariu, Weak bipolarizable graphs, Discrete Math., 74 (1989), 159-171.

[49] C. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.

[50] Various, *Perfect Graphs*, (J. Ramirez-Alfonsin, B. Reed, eds.), J.H. Wiley and Sons, Chichester U.K., (2001).

[51] T. Raschle, *Generalized Modular Decomposition and the Recognition of Classes of Perfectly Orderable Graphs*, Ph.D. thesis, Swiss federal institute of Technology, Zurich (1999).

[52] F. Rousell and I. Rusu, Holes and dominoes in Meyniel graphs, Internat. J. Found. Comput. Sci. 10 (1999) 127-146.

[53] D. J. Rose, R. E. Tarjan and G. S. Lueker, Algorithmic aspects of vertex elimination on graph, SIAM J. Compt., 5 (1976), 266-283.

[54] I. Rusu and J.P. Spinrad, Domination graphs: examples and counterexamples, Disc. Applied Math., 110 (2001), 289-300.

[55] D.P. Sumner, Graphs indecomposable with respect to the X-join, Discrete Math., 6 (1973), 281-298.

[56] J.P. Spinrad, P_4-trees and substitution decomposition, Discrete Appl. Math., 39 (1992), 263-291.

[57] J.P. Spinrad and R. Sritharan, Algorithms for weakly triangulated graphs, Discrete Appl. Math., 59 (1995), 181-191.

[58] R.I. Tyshkevich, Once more on matrogenic graphs, Discrete Math., 51 (1984), 91-100.

[59] D.J.A. Welsh and M.B. Powell, An upper bound on the chromatic number of a graph and its applications to timetabling problems, Compt. J., 10 (1967), 85-87.

[60] D.B. West, *Introduction to Graph Theory*, Prentice Hall, 2nd edition, (2001).

[61] I. E. Zverovich and I. I. Zverovich, The reducing copath method for simple homogeneous sets, Rutcor Research Report, 19 (2001)