Technische Universität Berlin
Fakultät II – Mathematik und Naturwissenschaften

Hook Graphs and More: Some Contributions to Geometric Graph Theory

Masterarbeit
im Studiengang Mathematik

Vorgelegt von	**Thomas Stuart Hixon**
Matrikelnummer	**319892**
Erstkorrektor	**Prof. Stefan Felsner**
Zweitkorrektor	**Prof. Rolf H. Möhring**
Abgabetermin	**Berlin, 25.05.13**

Die selbständige und eigenhändige Anfertigung
versichere ich an Eides statt.

Datum/ Ort/ Unterschrift

Abstract

In this work we focus on some problems in geometric graph theory. There are two main topics that we study. The main body (Chapter 2) of this thesis is about intersection graphs of geometric objects in the plane. The work in Chapter 2 was initially motivated by the study of segment graphs. A graph is a *segment graph* if the vertices can be represented by straight line segments in the plane, and (v, w) is and edge if and only if the two corresponding segments intersect. We introduce a class of segment graphs called cyclic segment graphs, where a graph is a *cyclic segment graph* if it has a segment representation whose segments all lie on lines tangent to a parabola and no two segments are parallel. We give various models of these graphs and show that bipartite cyclic segment graphs are exactly grid intersection graphs. A consequence of this, due to a result by Kratochvíl [36], is that it is NP-complete to test whether a graph is a cyclic segment graph. In later sections we discuss a subclass of cyclic segment graphs, which we call hook graphs. Hook graphs have also been studied independently by Cantanzaro et al. [9], who introduce hook graphs using a different model and are motivated by a problem in Biology. A *hook graph* is a segment graph that has a representation whose segments all lie tangent to a parabola and no two segments are parallel. In this thesis we introduce hook graphs using a different model of cyclic segment graphs. From this other model, we also prove that a graph is a hook graph if and only if it is the intersection graph of a set of axis aligned rectangles in the plane such that the top left corner of each rectangle lies on a unique point on the the line $\{(x, x) : x \in \mathbb{R}\}$. We also characterise hook graphs as being the graphs for which there exists an ordering of the vertices that satisfies a condition called the cross completion property. We use these various models to show that interval graphs, outerplanar graphs, and 2-directional orthogonal ray graphs are all hook graphs. In the final section on this topic, we give polynomial time algorithms to compute the clique number and the independence number of a hook graph. We also show that for any hook graph G we have $\chi(G) = O(\log(\omega(G)))$, where $\chi(G)$ and $\omega(G)$ are the chromatic number of G and the clique number of G, respectively. This result follows from a theorem by Chalermsook [10] about rectangle intersection graphs. We conclude the topic by showing that the independence number is a 2-approximation of the clique covering number for all hook graphs.

The other topic in my thesis is the focus of Chapter 3. Here, we discuss a colouring problem that is related to conflict-free colourings of point sets. Namely, let $\mathcal{R}_L$ be the set of all axis aligned rectangles in $\mathbb{R}^2$ that intersect a fixed horizontal line L. We define $\chi_k(\mathcal{R}_L)$ to be the minimum number of colours so that for any point set P, it is possible to find a colouring of the points so that for each rectangle $R \in \mathcal{R}_L$ with $|R \bigcap P| \geq k$, there are two points in $P \bigcap R$ that are coloured differently to each other. We show that the upper bounds on $\chi_k(\mathcal{R}_L)$ given by Keszegh [34] are best possible for $k = 2, \ldots, 5$. Together with the results by Keszegh [34], the only number k for which $\chi(\mathcal{R}_L)$ remains unknown is when k is 6.

Contents

Preface

In this work I study graphs that are induced by simple geometric objects in the plane, such as rectangles or line segments in the plane. I would like to point out that to my knowledge, strong connections between the two main topics in this thesis are yet to have been found; however, in both chapters the proofs of problems often exploit their geometric nature. For this reason, this thesis has been written so that two main chapters, Chapter 2 and Chapter 3, can be understood independently of each other. I have decided to include both of these topics because I have spent much time engaged in them and have encountered interesting problems in both of them, whose solutions encompass a range of techniques. The basic structure of this thesis is as follows:

- Chapter 1 introduces some basic fundamental notions in graph theory and explains the notation that I use in this thesis. I also include a brief introduction and an overview of the results discussed in Chapters 2 and 3.

- Chapter 2 discusses intersection graphs. In particular, I introduce some subclasses of segment graphs, which I call hook graphs and cyclic segment graphs. These classes arose as a result of looking at the underlying line arrangement of segment graphs. I present different models of these classes and relate them to some other classes that have already been studied. I also discuss the chromatic number, clique number, independence number, and clique covering number of hook graphs and cyclic segment graphs, giving algorithms to compute them explicitly in some cases and approximation algorithms in other cases.

- Chapter 3 focuses on a colouring problem, which is related to conflict free colourings of point sets in the plane. In particular, I show that some bounds given in [34] are tight. Namely, bounds on the chromatic number of the Delauney graph (and some generalisations of Delauney graphs) of points in the plane with respect to axis aligned rectangles that all intersect a horizontal line.

Acknowledgements

I would like to express my deep gratitude to my supervisor Prof. Stefan Felsner for having guided and supported me throughout the writing of this thesis. I would also like to thank Prof. Rolf H. Möhring for taking his time and being interested in my work. I would like to offer my special thanks to Nieke Aerts and Udo Hoffmann for the great help that they provided me with during the final stages of writing up my thesis. I am also grateful to all the members of the Diskrete Mathematik group at the Technische Universität Berlin for many fruitful discussions, which helped me progress in my understanding of the topics. I wish to extend my thanks to Dr. George Mertzios for the discussions and support that he provided me with, which helped and motivated me. Finally, I wish to thank my family and friends for their encouragement and support during the completion of this thesis.

1. Introduction

Graph theory is an important field in mathematics, with applications in a range of fields including computer science, networking, and biology. This thesis focuses on graphs that are induced by geometric objects such as line segments and rectangles in the plane.

Given a graph $G = (V, E)$, there are many ways to represent G geometrically. The most common way to represent graphs is by representing the vertices $v \in V$ by points p_v in the plane and edges $e = (v, w)$ by smooth curves C_e whose endpoints are p_v and p_w. Usually, one also wants that the curve C_e does not go through other points p_x, for all vertices x that are not in e. Such a representation is called a *drawing* of a graph. Given a drawing of a graph, one can obtain the vertices and the edges of the graph from the picture, i.e., the drawing fully describes the graph in question. One often asks whether there is a drawing of a graph that satisfies additional properties, e.g., so that none of the curves C_e intersect each other unless they meet in a point p_v. A graph that can be drawn with this additional property is called planar. We discuss planar graphs in more detail in Section 1.2. Investigating classes of graphs is interesting because in many cases, problems that are difficult to solve for graphs in general may be easier to solve for certain graph classes, for example, finding a maximum clique in a planar graph can be done in a time that is polynomial in the number of vertices, but computing the size of a maximum clique for a graph in general is NP-hard.

In Chapter 2 we focus on classes of graphs that are called intersection graphs. Given a family $\mathcal{F}$ of sets, the graph G is called an *intersection graph* of $\mathcal{F}$ if for each vertex v of G, there is an element F_v in the set $\mathcal{F}$ so that $F_v \bigcap F_w \neq \emptyset$ if and only if (v, w) is an edge of G. We only focus on families $\mathcal{F}$, whose elements can easily be described geometrically. Geometric families $\mathcal{F}$ that have been studied so far include the set of axis-aligned rectangles in the plane, the set of segments in the plane, and the set of intervals on a line. All these intersection graphs will be discussed in more detail in Chapter 2. One nice feature of the intersection graphs mentioned above except segment graphs, is that one can store them using a memory space that is linear in the number of vertices of the graph. Segment graphs are more complicated in general as the amount of space needed to store the end points of the segments can require a memory space that is exponential in the number of vertices (see Kratochvíl and Matoušek [37]). The structure of segment graphs in general is not well understood. For this reason, in Section 2.1 we introduce a new subclass of segment graphs called cyclic segment graphs. Cyclic segment graphs are investigated as a consequence of looking at segment graphs whose segments lie on a specific line arrangement (up to homeomorphisms of the plane). More precisely, cyclic segments graphs are defined to be segment graphs whose segments lie on a set of lines tangent to a parabola. We give various models of this class and manipulate one of these models to show that bipartite cyclic segment graphs are exactly grid intersection graphs. A consequence of this result is that the recognition problem for cyclic segment graphs is NP-hard even if we restrict ourselves to bipartite graphs. In the last two sections of the Chapter 2, we look at a certain subclass of cyclic segment graphs, which we call hook graphs. These graphs have also been investigated independently by Cantanzaro et al. [9], who are motivated to study hook graphs by a problem in biology. In Section 2.2 we discuss various models of hook graphs and show that 2-directional orthogonal ray graphs, interval graphs and outerplanar graphs all have hook representations. We also note that hook graphs are a subclass of intersection graphs of axis-aligned rectangles in the plane. In Section 2.3 we investigate the perfect graph parameters of hook graphs, i.e., the chromatic number, the clique number, the clique covering number, and the independence number of hook graphs. We give polynomial time algorithms to find a maximum clique and a maximum independent set of a hook graph. Applying a result by Chalermsook [10], we also give an $O(\omega(G)\log(\omega(G)))$ bound on the chromatic number of a graph G, where $\omega(G)$ denotes the clique number of G. Improving the upper bound on the chromatic number of hook graphs is a particularly interesting question. For rectangle intersection graphs it is an unsolved and challenging question to find a tight upper bound on the chromatic number by a function of the clique number. The best known lower bound is linear and the upper bound is quadratic in $\omega(G)$. We also show that the complement of a cycle with more than 6 vertices is not a hook graph and we give a full description of the possible hook representations that a cycle has. These results are motivated by the strong perfect graph theorem (Chudnovsky et al. [16]), which states that a graph is perfect if and only if it has no odd holes and no odd antiholes (see Section 1.3.2). We conclude the section by showing that all triangle-free hook graphs are 4-colourable. This shows a difference to rectangle intersection graphs in general as there exists a triangle-free axis-aligned rectangle intersection graph whose chromatic number is 6 (Asplund and Grünbaum [2]). We also include a proof of a result by Pawlik et al. [49], who show that for each number n there exists a triangle-free segment graph G_n whose chromatic number is n.

We conclude Chapter 2 by proving that the independence number of hook graphs is a 2-approximation of the clique covering number of hook graphs. Given a hook representation of a graph G, we also provide a polynomial time algorithm to compute a clique covering of size at most $2\alpha(G)$, where $\alpha(G)$ denotes the independence number of G.

In Chapter 3, we discuss graphs and hypergraphs, whose vertices correspond to a point set and whose edges are induced by a family of regions. Namely, consider a set P of points in $\mathbb{R}^n$ and a set $\mathcal{F}$ of regions in $\mathbb{R}^n$. We can define the graph $G(P, \mathcal{F})$ as follows. The vertex set of $G(P, \mathcal{F})$ is the point set P. A set $e \subset P$ is an edge of $G(P, \mathcal{F})$ if and only if there exists a region $r \in \mathcal{F}$ such that $e = r \bigcap P$. Motivated by a frequency assignment problem, which we explain in Section 3.1, we are interested in investigating bounds on the chromatic number of $G(P, \mathcal{F})$ for a fixed family $\mathcal{F}$ and for any point set P of n points. In Chapter 3, we focus on graphs $G(P, \mathcal{R}_l)$, where $\mathcal{R}_l$ is the set of all axis aligned rectangles that intersect a fixed horizontal line l. In the case where $\mathcal{F}$ is the set of all axis-aligned rectangles in the plane, there does not exist an upper bound on the chromatic number of $G(P, \mathcal{F})$, that is, for any number n, one can find a point set P such that $G(P, \mathcal{F}) \geq k$. Keszegh [34] proved an upper bound of 6 on the chromatic number of $G(P, \mathcal{R}_l)$ for any finite point set P. We show that this bound is best possible by giving a point set P for which the chromatic number of $G(P, \mathcal{R}_l)$ is 6. Given a point set P in the plane, we also look at the graphs $G_k(P, \mathcal{R}_l)$, which have the same vertex set as the graph $G(P, \mathcal{R}_l)$, and a set e of vertices is an edge of $G_k(P, \mathcal{R}_l)$ if and only if it is an edge of $G(P, \mathcal{R}_l)$ that contains at least k vertices. We show that some bounds given by Keszegh [34] on $G_k(P, \mathcal{R}_l)$ are best possible. Given the results in this thesis, the only value of k for which it remains unknown whether the existing bound on $G_k(P, \mathcal{R}_l)$ is best possible is when k is 6.

More involved introductions, which include more motivation and history of the problems can be found within the chapters and sections in which the relevant problems are discussed. In the remainder of this chapter we introduce some notation and some fundamental concepts in basic graph theory, which we will use in later chapters.

1.1. Some Basic Definitions

The first thing we need to define is a graph.

Definition 1.1.1 (Graph)**.** A *graph* G is an ordered pair (V, E) where V is a set and E is a set of subsets of V of size 2. The elements of V are called *vertices*, and the elements of E are called *edges*. We denote an edge e consisting of 2 vertices u and v by (u, v). Unless stated otherwise, we let the natural numbers n and m denote $|V|$ and $|E|$, respectively.

Note that this is actually the definition of a simple graph because we have not allowed for loops and multiedges. Graphs can be used to model many things in the real world, for example a railway network, where the vertices are the stations and the edges are pairs of stations that have a connection between them with no stops in between. In Chapter 3 we will consider the following generalised definition, where edges may contain more than two vertices.

Definition 1.1.2 (Hypergraph)**.** A *hypergraph* is an ordered pair $(V, \mathcal{E})$ where V is the set of vertices and $\mathcal{E} \subset 2^V$ is the set of (hyper)edges, such that each edge contains at least two vertices. Here, 2^V denotes the power set of V.

In what follows, many of the definitions can be generalised to hypergraphs by replacing edges by hyperedges in the definitions. Hypergraphs only appear in Chapter 3 of this thesis. If the generalisation of a definition is used in Chapter 3 and the generalisation is not immediate, we will clarify it later. In what follows, we use the following notation:
- $[n] = \{1, 2, \ldots, n\}$ for a natural number n.
- Given a set A and a natural number $k \leq |A|$, then we let $\binom{A}{k} = \{B \subset A : |B| = k\}$.

Definition 1.1.3.

- Given two graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$, a function $h : V_1 \to V_2$ is called an *isomorphism* if h is a bijection and $(u, v) \in E_1 \iff (h(u), h(v)) \in E_2$. In this case, G and H are said to be *isomorphic*.
- A graph $H = (W, E_1)$ is called a *subgraph* of $G = (V, E)$ if $W \subset V$ and $E_1 \subset E$. Given a graph H, we say that G *contains* H if there is subgraph of G that is isomorphic to H.
- A graph $H = (W, E_1)$ is called an *induced subgraph* of G if $W \subset V$ and $E_1 = \{(u, v) \in E : u, v \in W\}$. We denote H by $G|_W$ and we sometimes called H the *restriction* of G on W.
- The *complement* of G is defined to be the graph $\overline{G} = (V, \overline{E})$, where $\overline{E} = \binom{V}{2} \setminus E$.

We now mention a few basic examples of graphs.

Definition 1.1.4. Given a graph $G = (V, E)$:

- $G = (V, E)$ is called a *path* if $V = \{v_1, v_2, \dots v_n\}$ and $E = \{(v_i, v_{i+1}) : i \leq n - 1\}$. Such a path is called a *path from* v_1 *to* v_n (see Figure 1.1). Given a path from v_1 to v_n, if we add the edge (v_1, v_n) then G is called a *cycle* (see Figure 1.2).
- G is called *connected* if for all $u, v \in V$ the graph G contains a path from u to v. Intuitively, this means that any vertex can be reached from another by going along edges.
- G is called a *forest* if it does not contain a cycle. If in addition, G is connected, then it is called a *tree* (see Figure 1.3).
- The *complete graph* on n vertices is defined by the graph $G = ([n], \binom{[n]}{2})$, which we denote by K_n.
- Maximal connected subgraph are called a *connected component* of G. The connected components of G create a partition of the vertex set.

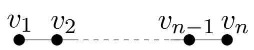


Figure 1.1.: A path.

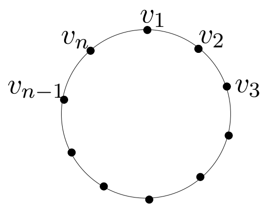


Figure 1.2.: A cycle.

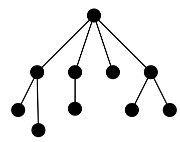


Figure 1.3.: A tree.

Definition 1.1.5 (Complete k-partite graph)**.** Given natural numbers $n_1, n_2, \dots, n_k$ we denote by $K_{n_1,n_2,\dots,n_k}$, the graph $G = (V, E)$ that has a partition of the vertices into sets $V_1, V_2, \dots, V_k$ such that $|V_i| = n_i$ for all $i \leq k$ and $E = \{(v, w) : v \in V_i, w \in V_k \text{ and } i \neq k\}$. The graph $K_{n_1,n_2,\dots,n_k}$ is called a *complete k-partite graph*; in the case that $k = 2$, the graph K_{n_1,n_2} is also called a complete *bipartite* graph. A graph is called *k-partite* if it is isomporphic to a subgraph of a complete k-partite graph.

We conclude this subsection with a useful equation about the relation between the degrees of the vertices and the total number of edges.

Definition 1.1.6 (Degree, average degree, and maximum degree)**.** Given a vertex v such that $(u, v) \in E$, we call u a *neighbour* of v and say that u is *adjacent* to v. The *neighbourhood* of v, denoted $N(v)$, is the set of all neighbours of v. The *degree* of v, denoted $\deg(v)$, is the cardinality of $N(v)$. The *maximum degree* of G, denoted $\triangle(G)$, is equal to $\max\{\deg(v) : v \in V\}$.

Proposition 1.1.7 (The degree-sum formula)**.** *Given a graph $G = (V, E)$, we have*

$$2|E| = \sum_{v \in V} \deg(v). \tag{1.1}$$

Proof. We show this equation by double counting the number of ordered pairs (v, e) such that $v \in V$, $e \in E$ and $v \in e$. If we count the number of edges incident to each vertex, we get the term on the right hand side of Equation 1.1.7. Each edge contains 2 vertices, therefore by summing over all edges, we get the left hand side. □

1.2. Planar Graphs

We often see maps of railway networks, where a station v is represented by a point p_v, and an edge $e = (v, w)$ is represented by a curve whose end points are p_v and p_w. This is called a drawing of the graph in the plane. Certain properties of such a drawing make it more understandable for the reader. In this section we introduce good drawings of graphs and planar graphs together with some of their properties. For a more detailed introduction to planar graphs we refer the reader to the book by Diestel [19].

Definition 1.2.1. A *good drawing (or embedding)* of a graph $G = (V, E)$ in $\mathbb{R}^2$ is a map $\phi : G \to \mathbb{R}^2$, where ϕ maps each vertex to a point in the plane, and each edge to a continuously differentiable curve that does not self intersect. In addition, we require that two vertices are not mapped to the same point and an edge e is not drawn through the image of a vertex that is not in e. Formally

- $\phi(u) \neq \phi(v)\ \forall v \neq u$, and
- given an edge (u, v), then $\phi((u, v)) \subset \mathbb{R}^2 \setminus \phi(V \setminus \{u, v\})$, with $\phi(u)$ and $\phi(v)$ as end points.

When talking about a drawing we will refer to a curve $\phi(e)$ and a point $\phi(v)$ by the edge e and the vertex v, respectively. It is easy to see that every graph has a good drawing in $\mathbb{R}^2$ as we can map the vertices to a set of points that are in general position (no three lie on a straight line) and we can define $\phi(e)$ to be the line segment between p_v and p_w, where $e = (v, w)$. Therefore any set of points in general position can give a good drawing of the graph. In the case of the railway network, we are interested in drawing graphs which are easier to read, or minimize certain criteria. 'Easier to read' is a very general statement; in graph theory, we formalise it in different ways depending on what we think 'easier' means. One example of this is to avoid crossings.

Definition 1.2.2 (Planar graph)**.** A graph G is a *planar graph* if there exist a (good) drawing of G such that if two curves $\phi(e)$ and $\phi(e')$ intersect then they must intersect in the image of a vertex. Such a drawing is called a *planar embedding* or a *plane graph.* Unless stated otherwise, we assume that an embedding of a planar graph is a planar embedding.

The Jordan Curve Theorem states that if we have a continuously differentiable loop, then removing it from the plane leaves two path connected regions, one of which is bounded, the other unbounded. Using the Jordan Curve Theorem, we see that when removing the planar embedding from the plane, we are left with a collection of bounded regions and one unbounded region. The bounded regions are called *bounded faces* and the unbounded region is called the *unbounded face.* Given a plane graph $G = (V, E)$ we define the *dual* of G to be $G^* = (V^*, E^*)$, where V^* corresponds to the set of faces of G. The pair (f_1, f_2) is in E^* if and only if there exists an edge e in E such that e lies on the boundaries of f_1 and f_2. Note that f_1 and f_2 are not necessarily different faces, and that the dual graph of a simple plane graph is not necessarily simple.

Remark 1.2.3. *Given a planar graph, then* 2 *different embeddings could have duals which are not isomorphic. Figure 1.4 shows two embeddings of a graph (shown with black vertices) together with its dual (shown with white vertices). The dual of the left embedding has* 2 *vertices that have* 5 *neighbours, whereas the dual of the right embedding only has one vertex with* 5 *neighbours. This shows that in general the dual is only well defined for a given embedding.*

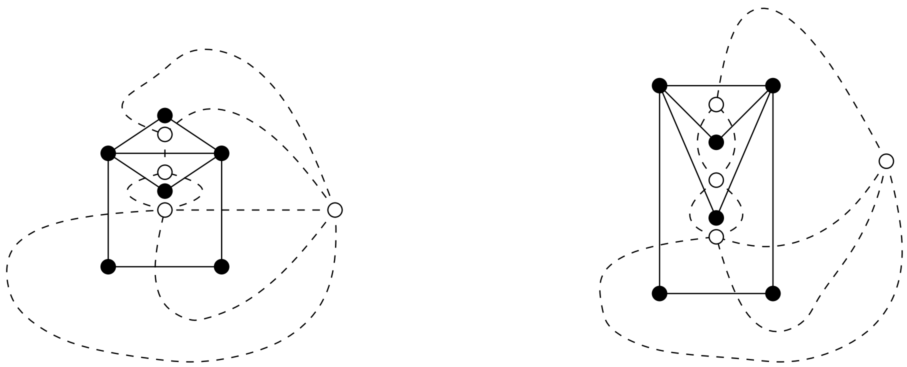


Figure 1.4.: Two planar embeddings of a graph with non isomorphic duals.

Definition 1.2.4. A *triangulation* T is a maximal (simple) plane graph with at least 3 vertices, i.e., if an edge is added to the edge set then it is no longer planar.

One property of triangulations is that the (edge) boundary of each face is of length 3 and consists of exactly three different edges. Another property is that every triangulation on more than three vertices has a simple dual graph whose vertices all have degree equal to 3.

Definition 1.2.5. An *outerplanar graph* is a planar graph that has an embedding G with dual G^* such that the vertex in V^* corresponding to the outerface of G is adjacent to all other vertices in V^*.

Although the dual of a planar graph may not be unique, the following result implies that the number of faces in an embedding of a planar graph does not depend on the embedding.

Theorem 1.2.6 (The Euler Formula (see [23]). *Let G be a (simple) plane graph, whose set of vertices, edges, and faces are V, E, and F, respectively. Then, we have*

$$|V| - |E| + |F| = 2.$$

The following bounds can be proved by applying the Euler Formula and double counting the number of pairs (e, f) such that e is an edge that is on the boundary of f.

Remark 1.2.7. *For $G = (V, E)$ a plane graph with face set F, the following hold:*

- $|E| \leq 3|V| - 6$

- $|F| \leq 2|V| - 4$
- *If G contains no cycle of length 3, then* $|E| \leq 2|V| - 4$

Using these upper bounds, it is not hard to show that K_5 and $K_{3,3}$ are not planar. We can also conclude that planar graphs are not very dense, i.e., planar graphs do not have many edges. Kuratowski [40] showed a characterisation of graphs that are not planar. For a proof that is written in English, we refer the reader to the book by Diestel [19]. To understand the statement of the theorem, we need the following definition.

Definition 1.2.8. We say we *subdivide an edge* $e = (u, v)$ when we introduce a new vertex u' to the vertex set and replace e by the two edges (u, u') and (u', v) (see Figure 1.5). A *subdivision of a graph* G is a graph G' that is obtained by subdiving edges of the graph G, i.e., there exists a sequence of graphs $G = G_1, G_2, \ldots, G_n = G'$ such that G_i is obtained by subdividing an edge in G_{i-1}.

An edge e. $\qquad$ e subdivided.

Figure 1.5.: The subdivision of an edge.

Theorem 1.2.9 (Kuratowski)**.** *A graph G is planar if and only if no induced subgraph of G is isomorphic to a subdivision of $K_{3,3}$, and K_5.*

This theorem implies that being a planar graph can be defined in a purely combinatorial way. One final result we should mention is the famous Four Colour Theorem, which was an open problem for over a century. The result was first proved with the aid of a computer in 1976 by Appel and Haken [30]. This proof however, is unable to be checked by hand and was not accepted by everyone when it was published. In 1997, Robertson et al. [51] published a proof that is easier to check, but is still computer aided.

Theorem 1.2.10 (The Four Colour Theorem)**.** *The chromatic number (see Definition 1.3.1) of planar graphs is at most* 4.

1.3. The 4 Perfect Graph Parameters

Here we discuss the 4 perfect graph parameters whose computation have many practical applications. Bounds on these parameters, their computational complexity, and possible approximation algorithms are some of the many interesting problems that are investigated when considering different graph classes.

Definition 1.3.1 (Good colourings and the chromatic number $\chi(G)$)**.** Given a graph (or hypergraph) G, a *good colouring* of G is a map c from the vertices to a finite set $\mathcal{C}$ such that for each edge e there exists two vertices $u, v \in e$ with $c(u) \neq c(v)$. An edge that does not satisfy this condition is called *monochromatic*. The elements of C are called the *colours* and we call $c(v)$ the *colour* of v. Given a colour $c_i \in \mathcal{C}$, the set $\{v \in V : c(v) = c_i\}$ is called a *colour class*. If the cardinality of $\mathcal{C}$ is k, then we say that G is k-colourable.

- The *chromatic number* of G, denoted $\chi(G)$, is the minimum value of k for which G is k-colourable.

Remark 1.3.2. *If a graph G is k-partite, then $\chi(G) \leq k$.*

The following is an example of a practical application of the chromatic number. Suppose the head of a company wants to organise some meetings, where all members of the company should attend one meeting. For a meeting to go well, he would like to avoid that one person does not like another in the meeting. To this end, we define a conflict graph of a group of people to be the graph $G = (V, E)$, whose vertices correspond to people, and whose edges are pairs of people where at least one of them dislikes the other. The chromatic number of this conflict graph of employees would then correspond to the minimum number of meetings needed.

Definition 1.3.3 (Clique number $\omega(G)$)**.** Let $G = (V, E)$ be a graph, then:

- A set $C \subset V$ is called a *clique* if $G|_C$ is isomorphic to the complete graph $K_{|C|}$.

- The *clique number* of G, denoted $\omega(G)$, is the size of the largest clique in G (denoted $\omega(G)$). We call a graph G *triangle-free* if there is no subgraph of G that is isomorphic to K_3.

Remark 1.3.4. *Given a graph G, we have that $\omega(G) \leq \chi(G)$. This is because no two vertices in the same clique can have the same colour in a good colouring of G.*

Definition 1.3.5 (Independence number $\alpha(G)$)**.** A subset of the vertices $\mathcal{I}$ is called an *independent set* if $G|_{\mathcal{I}}$ has no edges. The size of the largest independent set in G is called the *independence number* of G, denoted $\alpha(G)$.

The independence number of the conflict graph of employees as defined above, gives the biggest possible size of a good meeting.

Remark 1.3.6 ($\alpha(G) \geq \frac{n}{\chi(G)}$)**.** *Let ϕ be a good colouring of a graph G. Then a colour class of ϕ is an independent set. Using this, we have that $\alpha(G)$ is at least as big as the largest colour class of ϕ, which at least as big as the average size of a colour class of ϕ. Taking a good colouring with $\chi(G)$ colours, we get $\alpha(G) \geq \frac{n}{\chi(G)}$.*

Definition 1.3.7 (Clique covering number $\gamma(G)$)**.** A *clique covering* of a graph G, is a partition of the vertex set into cliques. The *clique covering number* of G, denoted $\gamma(G)$, is the minimum size of a clique covering of G.

Remark 1.3.8. *A clique in G is an independent set in $\overline{G}$, and therefore a clique covering of G corresponds to a partition of V into colour classes of a good colouring of $\overline{G}$. From this we obtain:*

- $\omega(G) = \alpha(\overline{G})$.

- $\gamma(G) = \chi(\overline{G})$.

- $\gamma(G) \geq \alpha(G)$, *from remark 1.3.4.*

Remark 1.3.9 ($\chi(G) \leq \triangle(G)+1$)**.** *A good colouring using $\triangle(G)+1$ colours can be obtained by colouring the vertices one by one maintaining that no edge is monochromatic. Because we have $\triangle(G)+1$ colours available, we know that for each vertex v there exists one colour that has not been used in its neighbourhood, which we assign to $c(v)$.*

Definition 1.3.10. A graph G is *k-degenerate* if there exists an ordering of the vertices $v_1, v_2, \ldots, v_n$ such that $\deg(v_i)$ is less than $k+1$ in $G|_{V_i}$, where V_i is the set $\{v_1, v_2, \ldots, v_k\}$.

Proposition 1.3.11. *Every k-degenerate graph is $k+1$ colourable.*

Proof. Let G be a k-degenerate graph. We colour the vertices in a k-degenerate order. When colouring a vertex, we use a colour that has not been used on any of its neighbours that have already been coloured. We can always find a free colour because when we colour a vertex it is only adjacent to at most k vertices that have already been coloured. Therefore, there is always a colour available for this vertex. It follows that we have a good colouring of G. □

Proposition 1.3.12. *A (finite) forest is 1-degenerate.*

Proof. Given a forest G, then consider a vertex v_1 of G. If v_1 is a *leaf*, i.e., $\deg(v_1) = 1$, then we are done. If v_1 is not a leaf, then let v_2 be a neighbour of v_1. Continuing like this, consider the vertex v_i, for $i \geq 2$; if v_i is not a leaf, then let v_{i+1} be a neighbour of v_i that is not v_{i-1}. Because we have a tree, we have that $v_i \neq v_j$ for $i \neq j$, otherwise G would contain a cycle. Therefore as our graph is finite, this process must end and the last vertex that we have encountered is a leaf. Removing this vertex, we obtain another forest for which we can find another leaf in the same manner. By reversing the order of the vertices that we removed, we obtain an order of vertices that satisfies the 1-degeneracy condition. □

From the discussion above, we get the following:

Corollary 1.3.13. *Trees are 2 colourable.*

1.3.1. Algorithms

Given an algorithm, its *(run time) complexity* is the maximum time that it takes to run through the algorithm given an input of size n. The complexity is written as a function of the input size. The function may also be dependent on other parameters. A problem is said to be solvable in linear (resp. polynomial, or exponential) time, if there exists an algorithm whose complexity is a linear (resp. polynomial, or exponential) function of n. The complexity of an algorithm is often written using the Θ notation, which ignores coefficients and lower order terms, for example, for the function $f(n) = 5n^4 + 2n^2 + 1$, one can write $f(n) = \Theta(n^4)$.

Let f be a function and let $g_1(n)$ and $g_2(n)$ be two other functions such that $g_1(n) \leq f(n) \leq g_2(n)$ for all n. Then we can write $f(n) = \Omega(g_2(n))$ and $f(n) = O(g_2(n))$, where coefficients and lower order terms are also ignored when writing $\Omega(g_2(n))$ and $O(g_2(n))$, for example, if $n^2 - 5 \leq f(x) \leq 3n^4 + n^3$ then $f(n) = \Omega(n^2)$ and $f(n) = O(n^4)$. If in addition, $\Theta(g_1(n)) = \Theta(g_2(n)) = g(n)$, we can conclude that $f(n) = \Theta(g(n))$. This can be useful if we have a complicated function $f(n)$ or a function that we cannot state explicitly.

Definition 1.3.14.

- A *decision problem* is a problem whose answer is either 'yes' or 'no'. For a fixed number $k \geq 3$, determining whether a graph is k-colourable is a decision problem.
- A *(combinatorial) optimisation problem* is a problem where we want the best possible solution with respect to a given order. Finding the chromatic number of a graph is an example of an optimisation problem.
- A problem is said to be in *NP*, if it is a decision problem and there exists a certificate such that one can check this certificate in polynomial time with respect to the size of the input in the question. A *certificate* in this case is a correct solution if the answer is yes, for example, the question whether a graph is 3-colourable is in NP: If a graph is 3-colourable, then we can choose the certificate to return a good 3-colouring of the graph, which can be verified in quadratic time whether it is good.
- Given a problem Q we say that we can *reduce* it to a problem P if we can find a polynomial time algorithm to solve problem P that uses an algorithm to solve problem Q as its only subroutine.
- A problem Q is said to be *NP-hard* if it is as hard as the hardest problem in NP, that is, we can reduce all problems Q_p in NP to problem Q.
- P is said to be *NP-complete* if it belongs to NP and is NP-hard.

Note that if we reduce a NP-hard problem to another problem, then this other problem is also NP-hard. It was shown by Cook [17] that a problem called 3-SAT is NP-complete. Since then, 3-SAT has been reduced to many other problems, which in turn were often reduced to other problems. Proofs that show NP-hardness usually find a reduction from a problem that is known to be NP-hard. An example of such a reduction will be given in Chapter 2, where we show that the recognition problem for cyclic segment graphs is NP-complete. The *recognition problem* for a class $\mathbf{C}$ of graphs is defined to be the problem of checking whether a graph G is a member of $\mathbf{C}$ or not. The main motivation for introducing NP-hardness is that there are many problems in NP for which no polynomial time algorithm has been found to solve any of them. However, it remains unsolved and is one of the millenium problems whether P = NP. By definition, it would be enough to find a polynomial time algorithm to solve one of the NP-complete problems, in order to find a polynomial time algorithm for each NP-complete problem.

Approximation algorithms Often, we cannot find a polynomial time algorithm to solve a problem. When dealing with optimisation problems one can try to find another algorithm which approximates the correct solution. Such an algorithm is called an *approximation algorithm.*

Definition 1.3.15 (ρ-approximation)**.** Let $f(G)$ and $g(G)$ be graph parameters, such that $f(G)$ and $g(G)$ are real numbers. If $g(G) \leq f(G) \leq \rho g(G)$ for all graphs, then $g(G)$ is called a ρ-approximation of $f(G)$. Note that ρ can be a function of $g(G)$.

As mentioned above, one might not be able to find a fast algorithm to compute $f(G)$. In this case, minimising ρ for which we can find an approximation algorithm with a fast complexity could be useful. In Chapter 2, we will show that the independence number $\alpha(G)$ is a 2-approximation of the clique covering number $\gamma(G)$ for hook graphs. We also find a quadratic time algorithm to compute the clique number of hook graphs given a representation. In general, it is not true that $\gamma(G)$ is bounded by a function of $\alpha(G)$: Mycielski [44] showed a construction of triangle-free graphs with arbitrarily high chromatic number. Taking the complement of those graphs, we obtain graphs with maximal independent set of size 2 and arbitrarily large clique covering number. In Chapter 2 we give a sketch of a construction by Pawlik et al. [49] that gives triangle-free segment intersection graphs with arbitrarily large chromatic number. If G is an intersection graphs of axis alligned rectangles, Asplund and Grünbaum [2] proved that $\chi(G) = O((\omega)^2)$ and they asked whether a linear bound exists. The best known bound by Hendler [32] remains quadratic. However, some improvements have been made by Chalermsook [10] for rectangle representations that have no containment intersections. As hook graphs have a representation without containment intersections, we apply the result by Chalermsook [10] to show that the chromatic number of hook graphs is of size $O(\omega(G)\log(\omega(G)))$.

1.3.2. Perfect Graphs

We conclude this chapter by stating a few results about perfect graphs and discussing some graph classes that are perfect. For a better understanding of perfect graphs, we refer the reader to the book by Dienstel [19]. As mentioned above, it can be interesting to study how the clique number and the chromatic number of graphs in a certain graph class relate to each other. One can also study the properties of graphs with a chromatic number and a clique number that relate in a certain way. One class of graphs that is defined in this way is the class of perfect graphs.

Definition 1.3.16. A graph G is called *perfect* if for all induced subgraphs G' of G, we have $\chi(G') = \omega(G')$.

Perfect graphs were introduced by Berge [6], who also made 2 famous conjectures in 1961. The first one was proved by Lovasz [41], which states that the complement of perfect graphs are perfect. After much interest, the second conjecture was proved by Chudnovsky et al. [16], who published it in 2006. Their proof is based on much research in the structure of Berge graphs (defined below).

Theorem 1.3.17 (The strong perfect graph theorem)**.** *A graph G is perfect if and only if it does not contain an odd hole or an odd antihole, where:*

- *An odd hole is an induced subgraph that is isomorphic to C_{2k+1} for a given $k \geq 2$.*
- *An odd antihole is an induced subgraph that is isomorphic to $\overline{C_{2k+1}}$ for a given $k \geq 2$.*

Graphs that don't have odd holes or antiholes are called Berge graphs*.*

From this characterisation by Berge graphs, one can easily deduce that chordal graphs are perfect.

Definition 1.3.18. A graph G is called *chordal* if there is no number $k \geq 4$ for which the cycle on k vertices C_k is an induced subgraph of G.

The next remark implies that chordal graphs are perfect.

Remark 1.3.19. *Chordal graphs do not contain any odd antiholes.*

Proof. The graph C_5 is isomorphic to $\overline{C_5}$, therefore it is not contained in a chordal graph. It is not hard to see that $\overline{C_k}$ contains an induced C_4 for all $k \geq 6$, therefore $\overline{C_k}$ is not chordal for $k \geq 5$. Being chordal is heredetary, therefore we have that chordal graphs contain no odd antiholes. $\square$

Another class of graphs that are perfect are comparibility graphs. In order to define a comparability graph, we need to define a poset.

Definition 1.3.20 (poset)**.** A *partially ordered set (poset)* is an ordered pair $(P, \leq)$, where P is a set and $\leq$ is a binary relation that satisfies the following properties:

- Reflexivity: For all $x \in P$ we have $x \leq x$.
- Antisymmetry: If $x \leq y$ and $y \leq x$, then we have $x = y$.
- Transitivity: If $x \leq y \leq z$, then we have $x \leq z$.

A binary relation that satisfies these three properties is called a *partial order*.

Definition 1.3.21. A graph $G = (V, E)$ is called a *comparability graph* if we can define a partial order $\leq_v$ on the vertex set V, so that for all vertices v_1 and v_2 with $v_1 \neq v_2$ we have:

$$(v_1, v_2) \in E \text{ if and only if } v_1 \leq_v v_2 \text{ or } v_2 \leq v_1.$$

It is not too difficult to see that a comparability graph G cannot contain an odd hole or an odd antihole. The key tool that one can use to prove this is as follows:
If (v_1, v_3) and (v_2, v_3) are both edges of G, but (v_1, v_2) is not an edge of G, then from transitivity we have $v_1 \leq v_3$ if and only if $v_2 \leq v_3$.

There are many other classes of graphs that are perfect; however, we only decided to include these classes here because they will be mentioned later in this thesis.

2. Cyclic Segment Graphs and Diagonal Hook Graphs

In this chapter we investigate problems on certain classes of geometric intersection graphs. More specifically, we introduce two new classes of intersection graphs, which we call cyclic segment graphs and hook graphs. We study some of their properties and discuss relations to other classes of intersection graphs that appear in the literature. We include various models of the two graph classes, which we use to tackle many of the problems that we discuss.

In Section 2.1 we introduce cyclic segment graphs and discuss related graph classes that initially motivated us to study cyclic segment graphs. We give various models of cyclic segment graphs and use these models to show that bipartite hook graphs are exactly grid intersection graphs. A consequence of this is that the recognition problem for cyclic segment graphs is NP-complete. In Section 2.2 we look at a subclass of cyclic segment graphs, which we call hook graphs. We characterise hook graphs as being graphs for which there exists an ordering of the vertices that satisfies a property that we call the 'cross completion property'. We also show that outerplanar graphs, 2-directional orthogonal ray graphs, and interval graphs all have hook representations. In Section 2.3 we give polynomial time algorithms to compute a maximum-weight clique and a maximum-weight independent set of a hook graph given a weight function on the vertices. We also give polynomial time approximation algorithms to compute the clique covering number and the chromatic number of hook graphs. These approximation algorithms use the fact that hook graphs can be represented as rectangle intersection graphs in the plane.

2.1. Cyclic Segment Graphs

We begin by giving the definition of an intersection graph.

Definition 2.1.1. Given a family of subsets $\mathbf{F}$ of a set X, G is called an *intersection graph* of $\mathbf{F}$ if there is an injective map $f : (V, E) \to \mathbf{F}$, such that for $v, w \in V$

$$(v, w) \in E \iff f(v) \bigcap f(w) \neq \emptyset.$$

The first thing to notice is that every graph is an intersection graph. Indeed, given a graph $G = (V, E)$, we can assign to each vertex $v \in V$, the set $F(v) = \{(v, w) : (v, w) \in E\}$. We now show that G is an intersection graph of $\mathbf{F} = \{F(v) \ : \ v \ \in V \ \}$. This is true because $F(v) \bigcap F(w) \neq \emptyset$ if and only if there is an edge $e \in E$ with $e \in F(v)$ and $e \in F(w)$, which is true if and only if $v \in e$ and $w \in e$, which is equivalent to $(v, w) \in E$.

In this thesis we focus on *geometric intersection graphs*, where $\mathbf{F}$ is a subset of a family $\mathcal{F}$ of geometric objects such as rectangles or disks in the plane. Such intersection graphs are called intersection graphs of $\mathcal{F}$.

Examples of geometric intersection graphs that have been the focus of much research are when:

- $\mathcal{F}$ is the set of intervals on the real line. Intersection graphs of intervals on the real line are also called *interval graphs*.
- $\mathcal{F}$ is the set of convex sets in the plane.
- $\mathcal{F}$ is the set of Jordan curves in the plane. A *Jordan curve in the plane* is defined to be the image of a continuously differentiable function $f : [0, 1] \to \mathbb{R}$. Intersection graphs of Jordan curves in the plane are also called *string graphs*.
- $\mathcal{F}$ is the set of line segments in the plane. Intersection graphs of line segments in the plane are also called *segment graphs*.

The specific choice of a set $\mathbf{F} \subset \mathcal{F}$ for a graph G is called a *representation* of G. A property P of a graph is called *hereditary* if given any graph G with property P, then all induced subgraphs of G also have property P. Being an intersection graph of a family $\mathcal{F}$ is hereditary. Indeed, given an intersection graph $G = (V, E)$ of a family $\mathcal{F}$ and a subset of the vertices H, then any representation $\mathbf{F}$ of G restricted to the elements of $\mathbf{F}$ that correspond to the vertices of H is a representation of $G|_H$.

2.1.1. Related Graph Classes

In this subsection, we discuss some results about the geometric intersection graphs mentioned above. We conclude this subsection with an open question that initially motivated us to investigate cyclic segment graphs.

Remark 2.1.2.
1. *G is an interval graph* $\implies$ **2**. *G is a segment graph* $\implies$ **3**. *G is the intersection of convex sets in the plane* $\implies$ **4**. *G is a string graph.*

Proof.
(1) $\implies$ (2) This is immediate as intervals on the real line can be seen as segments in $\mathbb{R}^2$.
(2) $\implies$ (3) This is also immediate because a segment is a convex set.
(3) $\implies$ (4) This can be seen by filling each convex set with a string, so that if two of the convex obects intersect then so do their corresponding strings (see Figure 2.1).

□

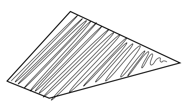
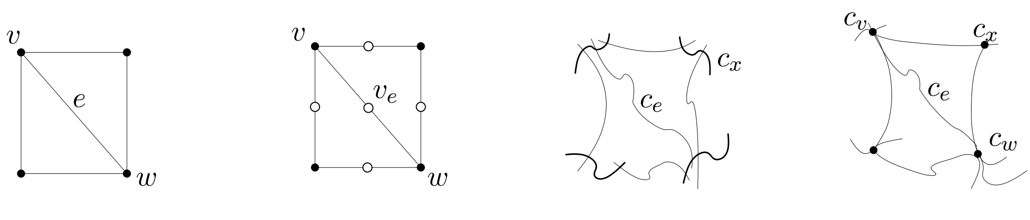


Figure 2.1.: Obtaining strings from convex regions in the plane.

The study of geometric intersection graphs began with the study of interval graphs, which were introduced by Benzer [4] in an article about genetic data analysis. Besides having real life applications in biology, interval graphs also have many applications in scheduling problems and printed circuit board designs. Since their introduction, there has been a lot of research about interval graphs and their structure is quite well understood. It is not difficult to see that interval graphs are chordal, therefore they are perfect graphs (see Subsection 1.3.2). Gilmore and Hoffman [26] showed that one can characterise interval graphs as chordal graphs whose complements are transitively orientable. They also showed that interval graphs are exactly those graphs whose maximal cliques can ordered so that for each vertex v, all the cliques that contain v are consecutive in the ordering. The latter has led to an $O(|V| + |E|)$ time algorithm by Booth and Lucker [7], who defined PQ-trees to solve the recognition question of whether a graph is an interval graph. Since then, other linear time algorithms that use lexicographic breadth first search and exploit other properties and characterisations of interval graphs have been constructed (see Corneil et al. [18], and Habib et al. [29]). One motivation of these more recent algorithms is to simplify the complexity analysis and to avoid the use of PQ-trees, which are not simple to program and involve many case distinctions. Regarding the perfect graph parameters, there are simple greedy linear time algorithms that compute a maximum clique, a maximum independent set, an optimal colouring, and a mimimum clique decomposition of interval graphs given an interval representations as input. For more about interval graphs we refer the reader to a book by Golumbic [27]. The study of intersection graphs has now become an broadly studied area of research. One direction in which interval graphs have been generalised is the study of segment graphs, string graphs and intersection graphs of other geometric objects in the plane. In this chapter we focus on subclasses of segment graphs and string graphs. We introduce some new graph classes, which give more insight into the structure of segment graphs. Before we introduce them, let us look at a few results from the literature to give an overview of the topic and motivate our results. We begin by showing that not all graphs are string graphs.

Proposition 2.1.3 (Ehrlich et al. [20])**.** *Full subdivisions of non planar graphs are not string graphs, where the* full subdivision *of a graph G is the graph obtained when subdividing every edge of G once.*



G $\quad$ $G^\bullet$ $\quad$ Representation $\quad$ Contraction

Figure 2.2.: Some steps in the proof of Proposition 2.1.3 shown on a **planar** graph G.

Proof. Let $G^\bullet$ be the full subdivision of a non-planar graph G and assume it is a string graph. Let V_S denote the set of vertices that have been added to G when subdividing (the white vertices in the figure above) and let V denote the set of vertices in $G^\bullet$ that are not in

V_S. The vertices in V form an independent set in $G^\bullet$. Therefore, in the string representation of $G^\bullet$, the set V corresponds to a collection of disjoint Jordan curves A in the plane. We can then contract each curve c_x in A to a point (see Figure 2.2) such that:

- The set of curves that each contracted curve intersects does not change.
- Uncontracted curves now intersect if and only if there exists a contracted curve c_x that they both intersected before the contraction, in which case they will intersect in the contraction of c_x.

Each curve c_e that has not been contracted corresponds to a subdivision vertex v_e, which in turn corresponds to an edge $e = (v, w)$ of G. The curve c_e intersects the contracted segments c_v and c_w of v and w, respectively. We remove any loops from c_e and 'cut' off the ends so that c_e has c_v and c_w as endpoints. We do this for all uncontracted curves c_e. None of the modified uncontracted curves intersect unless they meet in an endpoint. Therefore we have a planar drawing of G, which is a contradiction. □

Note that a subdivision of a graph is non planar by Kuratowski's Theorem (see Theorem 1.2.9), which states that a graph is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$. Ehrlich et al. [20] noticed that all planar graphs are string graphs: For each vertex, one can assign a path around the vertex which contains $\frac{3}{4}$ of each edge adjacent to that vertex as in Figure 2.3. The intersection graph of this set of strings is clearly isomorphic to the original graph. The dashed line in Figure 2.3 shows such a string for the vertex v.

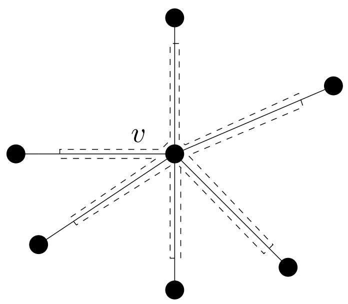


Figure 2.3.: Obtaining strings from a planar embedding.

One can also prove this result by applying Koebe's coin kissing theorem, which states that every planar graph is the intersection graph of discs in the plane such that no disc intersects the interior of another (for a proof, see the book by Pach and Agarwal [45]). One consequence of this approach is that we can also conclude that planar graphs have intersection representations of convex sets in the plane. One could ask the question whether there are string representations of planar graphs in which Jordan curves are allowed to pairwise intersect in at most **one** point in which they **cross**. Graphs with such string representations are called *pseudosegment intersection graphs* or 1-string graphs. In the constructions of string representations of planar graphs given above, the Jordan curves either cross more than once (in the first construction) or have intersection points that are not crossings (in the second construction). Scheinerman [53] conjectured in his PhD thesis, that planar graphs have a segment representation. After years of work on the topic by various researchers, Chalopin and Gonçalves [12] proved this result for pseudosegment intersection graphs, and later for segment graphs [13]. Regarding the complexity of recognising string graphs, Kratotchvíl [39] showed that the recognition of string graphs is NP-hard. It was unknown for a long time whether recognising string graphs was decidable, which was finally proved by Pach and Tóth [48]. One year later, it was shown by Schaefer et al. [52] that recognising string graphs is in NP, which means that recognising string graphs is NP-complete. Regarding the recognition of segment graphs, Kratochvíl and Matoušek [37] showed that the recognition of segment graphs has the same complexity as deciding truth in the existential theory of the reals, which is NP-hard and lies in PSPACE, but is unknown to be in NP. Kratochvíl and Matoušek [37] also show that some segment representations can need an exponential number of bits to store the end coordinates of the segments. They also show that for a number $k \geq 3$, the recognition problem is NP-complete for segment graphs whose segments are restricted to lie in at most k different directions (where parallel segments do not intersect). Kratotchvíl [36] showed that this result also holds for the case when $k = 2$.

It follows from the proof of Scheinerman's conjecture that the problem of computing the independence number, clique covering number, and the chromatic number of segment graphs is NP-hard, given the segment representation as an input. This is because all of these problems are NP-hard for planar graphs and because the construction of a segment representation of planar graphs given by Chalopin and Gonçalves [12] is found in polynomial time.

Recently, Cardinal et al. [8] showed that computing the clique number of a segment graph is NP-hard, by showing that for each planar graph G one can compute an even subdivision G' of G whose complement has a segment intersection representation. The important results that they prove are that G' is an even subdivision that can be computed in polynomial time, and that the segment representation of $\overline{G'}$ can be constructed in polynomial time. Using these results, they reduce the problem of computing the independence number of planar graphs to the problem of computing the clique number of segment graphs. The reduction is as follows: Given a planar graph G, compute the special subdivision G' whose complement is a segment graph. Then compute the segment representation of $\overline{G'}$. It follows that because G' is an even subdivision of G, the independence number of G can be computed in constant time given the clique number of $\overline{G'}$. Computing the independence number of planar graphs is NP-hard, therefore, computing the clique number of segment graphs is NP-hard. In fact, the representation that Cardinal et al. [8] find can be extended to a ray intersection representation. Therefore computing the clique number of ray intersection graphs is NP-hard.

From the results mentioned above, we see that many problems on segment graphs are much more difficult to solve than for interval graphs. Although many results about segment graphs are now known, their structure remains not very well understood.

Aside: An interesting open question A graph is called *coplanar* if it is the complement of a planar graph. If coplanar graphs are a subclass of segment graphs, then this would give a simpler reduction from computing the maximum independent set of planar graphs to computing the maximum clique of segment graphs. This problem, originally posed by Kratochvíl [38], remains unsolved and was the starting point of our research.

2.1.2. Models and the Recognition Problem

In this subsection we define cyclic segment graphs and prove the equivalence of various different models of these graphs. We conclude this subsection by showing that the recognition problem is NP-complete for cyclic segment graphs.

For the remainder of this thesis, unless stated otherwise, we consider segment graphs which have a representation such that no two segments are parallel. Given such a segment representation we can then extend the segments to lines in the plane, which we call the *underlying line arrangement* of the segment representation. For a better understanding of line arrangements see the book by Felsner [23]. As mentioned above, it is known to be difficult to recognise segment graphs in general. For this reason, we question whether the recognition problem is easier if we restrict our attention to segment graphs whose underlying arrangement is the cyclic arrangement (a set of lines that are tangent to a convex body). Normally, we consider line arrangements in the projective plane and two line arrangements are considered equivalent if there is a projective transformation that maps one line arrangement onto the other. For this reason, the line arrangement is well defined without specifying the convex body. It is not enough that two arrangements are projectively equivalent in order to conclude that the same segment graphs lie on them. Indeed, if we take the convex body to be a parabola, then it does not give the same class of graphs as if we were to take a closed convex body. The reason for this is that a projective transformation f might map a finite segment to a segment through infinity, although f preserves incidences and maps lines to lines. We will mention a specific difference between the classes later in this chapter. After some study, we realised that the segment graphs that lie on the set of lines tangent to a parabola have very interesting properties. This is how we came to investigate cyclic segment graphs, which we define as follows:

Definition 2.1.4. A segment graph G is called a *cyclic segment graph* if there exists a segment representation of G whose underlying line arrangement has all its lines tangent to a parabola.

Once again, we do not specify the parabola; however, this class is well defined as we can find a homeomorphism of the plane mapping the one parabola to the other. This homeomorphism also maps segments to segments in the Euclidean plane and preserves incidences.
Note that no two segments lie on the same line as we do not allow for parallel segments. For each vertex v, let S_v denote its corresponding segment and L_v denote the line on which S_v lies. Given a cyclic segment representation, we let $L_1, L_2, \ldots, L_n$ be the ordering of the lines so that the slope of L_i is less than the slope of L_j whenever $i < j$. We denote by S_i, the segment that lies on L_i in the representation. Let $v_1, v_2, \ldots, v_n$ be the ordering of the vertices such that v_i is associated to segment S_i in the representation; we call this ordering the *cyclic segment ordering* of the vertices.

We define the Θ-graph to be the graph shown in Figure 2.4. Figure 2.5 shows a cyclic segment representation of Θ together with the underlying arrangement, the ordering of the lines, and the parabola.

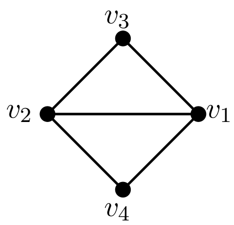


Figure 2.4.: The Θ-graph.

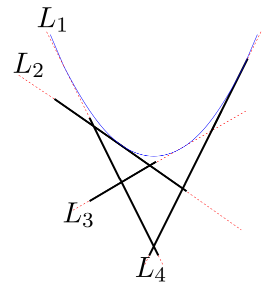


Figure 2.5.: Cyclic segment representation of Θ.

We now introduce another model, which we call the point-interval-containment model of the cyclic segment graph.

Definition 2.1.5. Given a graph $G = (V, E)$, we assign to each vertex v the ordered pair (I_v, p_v) where I_v is an interval in the real numbers and $p_v \in \mathbb{R}$ such that $p_v \neq p_w$ for $v \neq w$. We call the set $\{(I_v, p_v) : v \in V\}$ a *point-interval-containment representation* of G, or a *PIC representation)* of G if $(v, w) \in E$ if and only if $p_v \in I_w$ and $p_w \in I_v$.

Given a cyclic segment ordering $v_1, v_2, \ldots, v_n$, let p_i and I_i denote p_{v_i} and I_{v_i}, respectively.

Definition 2.1.6. A set of real intervals $I_1, I_2, \ldots, I_n$ is called a *indexed interval representation* of a graph $G = (V, E)$ if there exists an ordering of the vertices $v_1, v_2, \ldots, v_n$ such that (v_i, v_j) is an edge if and only if $i \in I_j$ and $j \in I_i$.

Proposition 2.1.7. *Given a graph $G = (V, E)$ then the following three statements are equivalent:*

- *G is a cyclic segment graph*
- *G has an indexed interval representation*
- *G has a PIC representation.*

Essentially, a point p_v in the PIC representation corresponds to an index in the indexed interval representation, which corresponds to a line L_v in the cyclic segment representation. An interval I_v corresponds to where a segment lies on L_v.

Proof. (1) $\implies$ (2) : Given a cyclic segment representation then let $v_1, v_2, \ldots, v_n$ be the cyclic segment ordering. We show that we can use this ordering for the indexed intervals, i.e., we assign to each v_i an interval I_i. The position of segment S_i on line L_i in the cyclic segment representation determines I_i. We use the points where other lines intersect I_i as a means to define where S_i lies on L_i. Namely, we label (on L_i) the intersection points between L_i and L_j with the number j. We also label the intersection point between L_i and the parabola with the number i (see Figure 2.6).

A consequence of the line ordering is that the labels on L_i are increasing from left to right. We let $I_i = [m_i - \frac{1}{2}, h_i + \frac{1}{2}]$, where m_i and h_i are taken to be the lowest and highest labels of intersection point that the segment S_i contains, respectively. In the special case that S_i doesn't contain a label, we let $I_i = [0, \frac{1}{2}]$ (note that in this case, the vertex v_i doesn't have

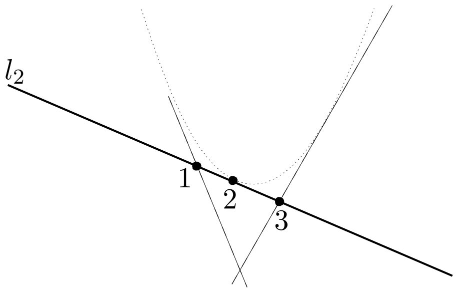


Figure 2.6.: Labelling the intersection points along line L_2.

neighbours). By construction (v_i, v_j) is an edge if and only if the intersection point labelled i (on L_j) is in S_j and the intersection point labelled j (on L_i) is in S_i, which is true if and only if $i \in I_j$ and $j \in I_i$. Therefore we have an indexed interval representation of G.
(2) $\implies$ (1) : Let $I_1, I_2, \ldots, I_n$ be an indexed interval representation of G and order the vertices $v_1, v_2, \ldots, v_n$ so that I_i corresponds to v_i. Draw n lines tangent to the parabola and label them $L_1, L_2, \ldots, L_n$ as above. We draw the segment S_{v_i} on on line L_i. Let j (resp. k) be the largest (respectively smallest) index of a neighbour of v_i, then we draw S_i so that one of its endpoints is the intersection between L_i with L_j, and the other is the intersection between L_i and L_k. We then extend all segments slightly to avoid segments of length zero and ensure that if two segments intersect then they cross. For an isolated vertex v_i, we define S_i to be a small segment on line L_i that touches the parabola and no other lines in the arrangement. The result is a cyclic segment representation of our graph.
(2) $\implies$ (3): If we have an indexed interval representation $I_1, I_2, \ldots, I_n$ of G then a PIC representation of G is $(I_1, 1), (I_2, 2), \ldots, (I_n, n)$.
(3) $\implies$ (2): Given a PIC representation $\{(I_v, p_v) : v \in V\}$ of G, let $v_1, v_2, \ldots, v_n$ be the ordering of the vertices such that $p_{v_i} < p_{v_j}$ when $i < j$. We claim that this is a valid ordering for the indexed intervals. To each non-isolated vertex v_i we assign the interval $I_i = [min\{j : p_{v_j} \in I_{v_i}\} - \frac{1}{2}, \max\{j : p_{v_j} \in I_{v_i}\} + \frac{1}{2}]$. We let I_i be the interval $[0, \frac{1}{2}]$ for any isolated vertex v_i. It is easy to see that $I_1, I_2, \ldots, I_n$ is an indexed interval representation of G. □

In the proof above, we saw that a cyclic segment ordering corresponds to an ordering of the vertices for the indexed interval representation, which corresponds to the ordering of the vertices with respect to the size of the p_v's. Applying the method in Proposition 2.1.7 to our representation of the Θ-graph (see Figure 2.5) we obtain the indexed intervals:

$$I_1 = [1.5, 4.5], I_2 = [0.5, 4.5], I_3 = [0.5, 2.5], \text{ and } I_4 = [0.5, 4.5].$$

Definition 2.1.8. Given a cyclic segment ordering of a graph G, we define the *canonical indexed interval representation* of G to be the set of indexed intervals $\{I_1, I_2, \ldots, I_n\}$, where I_i is defined as follows:

- $I_i = [\min\{j : (v_i, v_j) \in E\}, \max\{j : (v_i, v_j) \in E\}]$ if v_i has at least 2 neighbours.
- $I_i = [j - \frac{1}{2}, j + \frac{1}{2}]$ if v_i has only one neighbour v_j.
- $I_i = [i - \frac{1}{2}, i + \frac{1}{2}]$ if i has no neighbours.

This gives a quadratic time algorithm for computing an indexed interval representation given a cyclic segment ordering. Therefore, we can check in polynomial time if an ordering of the vertices of a graph is a valid cyclic segment ordering of the vertices by checking the indexed interval representation obtained above. The recognition of a cyclic segment graphs G is therefore in NP with a valid ordering as a certificate. Using the indexed interval model, we now show that cyclic segment graphs are a strict subclass of segment graphs.

Claim 2.1.9. *$K_{2,2,2}$ is not a cyclic segment graph.*

Proof. Given the tripartition V_1, V_2, V_3, assume $K_{2,2,2}$ is a cyclic segment graph. In this case, we can find an indexed interval representation, whose ordering is $v_1, v_2, \dots, v_n$. We restrict this representation to four vertices v_1, v_i, v_j, v_k, where i, j, and k are defined as follows.

- Assume $v_1 \in V_1$ and let v_k be the vertex with the highest index, which is not in V_1 (without loss of generality, let $v_k \in V_2$).
- Let v_i and v_j be the elements of V_3. By definition $1 < i, j < k$.

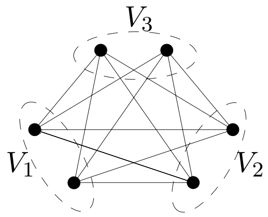


Figure 2.7.: $K_{2,2,2,}$.

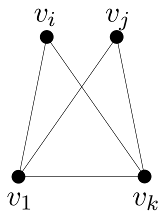


Figure 2.8.: Restriction to $\{v_1, v_i, v_j, v_k\}$.

We see that v_i and v_j are both connected to v_1 and v_k (see Figure 2.8), which means that I_i and I_j both contain 1 and k. This implies that $i \in I_j$, and $j \in I_i$ because $1 < i, j < k$. By definition of the indexed intervals representation we therefore get $(v_i, v_j) \in E$, which is a contradiction of the fact that (v_i, v_j) is not an edge of $K_{2,2,2}$. □

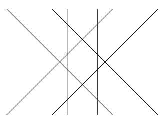


Figure 2.9.: A segment representation of $K_{2,2,2}$.

A segment representation of $K_{2,2,2}$ is shown in Figure 2.9. The underlying arrangement here is also the cyclic arrangement, so this shows it is necessary to specify that the underlying arrangement lies tangent to the parabola and not just any convex body. Having shown that cyclic segment graphs are a strict subset of segment graphs, one might hope to recognise cyclic segment graphs in polynomial time. However, the next theorem implies that a polynomial time algorithm that tests whether a graph G is a cyclic segment graph does not exist (unless P = NP).

Theorem 2.1.10. *Given a graph G, recognising whether G is a cyclic segment graph is NP-complete.*

We have already seen that recognising cyclic segment graphs is in NP. It remains to show that it is NP-hard, which is the aim of the remainder of this section. We begin by introducing another intersection model of cyclic segment graphs.

Definition 2.1.11. A *Generalised hook* is one of the following curves in the plane:

1. A vertical segment, which lies below the diagonal, i.e., the line $\{(x, y) \in \mathbb{R}^2 : x = y\}$.
2. A horizontal segment, which lies below the diagonal.
3. The union of a horizontal and a vertical segment which lie below the diagonal and intersect on the diagonal.

In all three cases above, the generalised hooks are allowed to touch the diagonal.
A *generalised hook graph* is the intersection graph of a set of generalised hooks that satisfies the following conditions:

- There are no intersections between parallel segments.
- There are no intersections between the horizontal (resp. vertical) parts of a hooks and horizontal (resp. vertical) segments.
- Generalised hooks don't intersect each other on the diagonal.

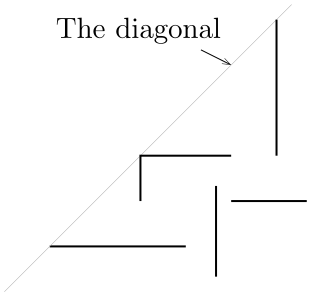


Figure 2.10.: A set of 5 disjoint generalised hooks.

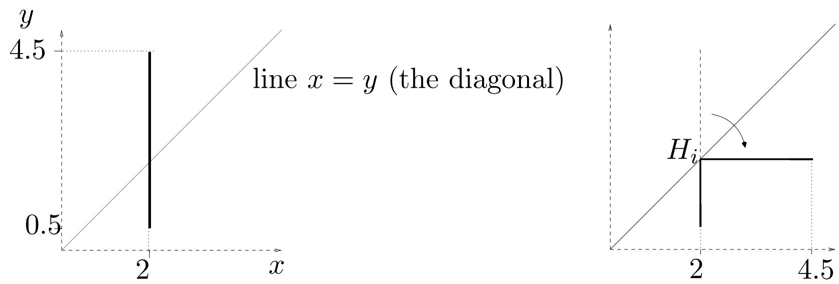
A generalised hook graph representation whose hooks all touch the diagonal is called a *hook representation.* A graph that has a hook representation is called a *hook graph.* Hook graphs will be discussed in detail later in the chapter.

Claim 2.1.12. *A graph G is a cyclic segment graph if and only if it is a generalised hook graph.*

Proof. Given a cyclic segment graph G with a PIC representation $\{(I_v, p_v) : v \in V\}$. We map each (I_v, p_v) to the vertical segment $\{p_v\} \times I_v$ in $\mathbb{R}^2$. We then reflect any part of this segment which lies above the diagonal to get the following set (see Figure 2.11):

$$H_v = \{(p_v, j); j \in I_v \text{ and } j \leq p_v\} \bigcup \{(j, p_v); j \in I_v \text{ and } j \geq p_v\}.$$

Figure 2.11 shows these steps for the ordered pair $([0.5, 4.5], 2)$. This delivers a set of generalised hooks whose intersection graph is isomorphic to G because H_v and H_w intersect if and only if $p_v \in I_w$ and $p_w \in I_v$.



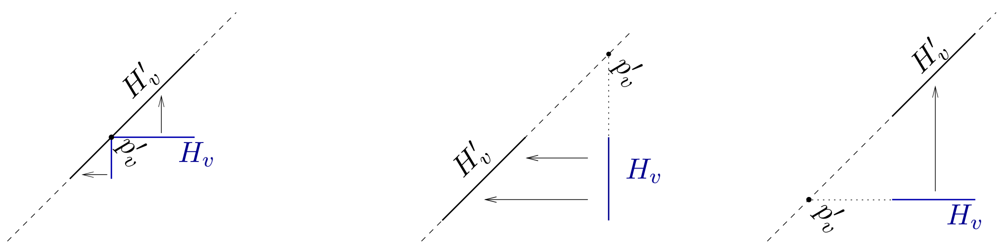
The vertical segment. The reflection in the diagonal.

Figure 2.11.: Reflecting a generalised hook in the diagonal.

For the other direction, take a generalised hook representation of a graph G. Reflect the horizontal part of each generalised hook H_v in the diagonal to obtain vertical segments of the form $p_v \times I_v$. The set $\{(I_v, p_v) : v \in V)\}$ is then a PIC representation of our graph after slightly perturbing any p_v's that coincide (without changing the point-interval-containments). $\square$

Definition 2.1.13. Given a cyclic segment ordering of G, let $\{I_1, I_2, \ldots, I_n\}$ be the corresponding canonical indexed interval representation. We define the *canonical generalised hook representation* of G to be the representation we obtain when following the steps of the proof above on the PIC representation $\{(I_1, 1)(I_2, 2), \ldots, (I_n, n)\}$.

Remark 2.1.14. *There is a nice visual way of obtaining the PIC representation from the generalised hook representation. Let p'_v be the intersection point between the generalised hook H_v and the diagonal. If H_v doesn't touch the diagonal then it must be a segment, in which case let p'_v be the intersection of the underlying line and the diagonal (see Figure 2.12). We then project each generalised hook H_v onto the diagonal obtaining a segment H'_v as shown by the arrows in Figure 2.12.*



Hooks touching the diagonal. Vertical hooks. Horizontal hooks.

Figure 2.12.: Mapping a hook to an interval and a point.

By construction, two generalised hooks H_v and H_w intersect if and only if points p'_w and p'_v are in H_v and H_w respectively. Considering the diagonal as the real line, the point p'_v can be seen as a real number p_v and the projected segment H'_v as an interval I_v. Therefore, $\{(H_v, p_v) : v \in V\}$ is a PIC representation of G.

Unless stated otherwise, given a generalised hook representation, we let $H_1, H_2, \ldots H_n$ be the ordering of the generalised hooks that corresponds to a cyclic segment ordering of the vertices, i.e., the ordering of their p_v's along the diagonal.

Definition 2.1.15. A graph is said to be a *grid intersection graph* if it is the intersection graph of horizontal and vertical segments in the plane such that parallel segments do not intersect.

Using the generalised hook model, we find a bijection between bipartite cyclic segment graphs and grid intersection graphs.

Proposition 2.1.16. *Finite bipartite cyclic segment graphs are exactly finite grid intersection graphs.*

Kratochvíl [36] proved that the recognition of grid intersection graphs is NP-complete. This result together with the fact that bipartite graphs can be recognised in polynomial time gives a proof of Theorem 2.1.10:

Proof of Theorem 2.1.10. Algorithm 1 shows the pseudocode of a reduction from the recognition of grid intersection graphs to the recognition of cyclic segment graphs. Note that it is important that the subroutine used to check whether a graph is bipartite is a polynomial time algorithm. The algorithm actually tells us if a graph is a bipartite cyclic segment graph or not, but this is the same as a grid intersection graph by the previous proposition.

Algorithm 1 GridIntReduction

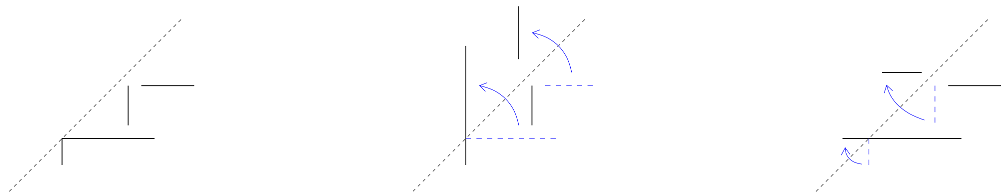
Input: A graph $G = (V, E)$
Output: 'YES' if G is a grid intersection graph, 'NO' if G is not a grid intersection graph.

if G is bipartite **then**
    if G is a cyclic segment graph **then**
        $ANSWER \leftarrow YES$
    else
        $ANSWER \leftarrow NO$
else
    $ANSWER \leftarrow NO$
return ANSWER

□

Proof of Proposition 2.1.2. For finite graphs we first show that grid intersection graphs are bipartite cyclic segment. A grid intersection representation of a finite graph must be contained in a bounded region. Therefore, this representation can be translated below diagonal to get a bipartite cyclic segment representation. Now for the other inclusion, consider a generalised hook representation of a bipartite cyclic segment graph $G = (V_1; V_2, E)$. Map the generalised hooks $H_i \in V_1$ to vertical segment H'_i by reflecting the horizontal parts of the generalised hooks in the diagonal (see Figure 2.13). If $H_i \in V_2$ we reflect only the vertical part in the diagonal to get H'_i. This maps V_1 and V_2 to sets of vertical and horizontal segments, respectively.

It is immediate from the following three observations that the intersection graph of these segments is isomorphic to G.



Set $\mathcal{H}$ of generalised hooks. The image if $\mathcal{H} \subset V_1$. The image if $\mathcal{H} \subset V_2$.

Figure 2.13.: The image of generalised hooks depending on their colour class.

1. There is no new intersection below the diagonal as no new part of a segment is created beneath the diagonal.

2. If there is an intersection between H'_i and H'_j above the diagonal, then H_i and H_j must intersect as anything that has been mapped above the diagonal must have been mapped there by the same map, namely, the reflection in the diagonal.

3. No intersection between H_i and H_j could have been removed: Either it was between a horizontal part of a vertex in V_1 and a vertical part of a vertex in V_2, in which case these lines were both reflected by the same mapping. Or vice versa, in which case the transformation would not have affected this intersection. Note that, no two of the parallel H'_i intersect.

Therefore bipartite cyclic segment graphs and grid intersection graphs are the same. □

This is an interesting result, which shows that two seemingly different classes are the same. We conclude this section by remarking that although grid intersection graphs are exactly bipartite cyclic segment graphs, claim 2.1.9 shows that not all segment graphs that have a representation using segments with three different slopes are cyclic segment graphs. Also note that $K_{2,2,2}$ is a planar graph, hence not all planar graphs are cyclic segment graphs.

2.2. Hook Graphs

In this section we investigate hook graphs and introduce a useful characterisation of hook graphs that is completely combinatorial. This characterisation is a tool that we often use later in this chapter. In Subsection 2.2.2, we compare hook graphs to other known graph classes. Although hook graphs seem quite restricted, it turns out that the class is rich in the sense that it contains other interesting and non trivial graph classes. We also discuss a special subclass of bipartite hook graphs, which we call stick graphs. We give an example of a graph that is a bipartite hook graph, but not a stick graph. We conclude this section by showing that hook graphs are a subclass of the class of intersection graphs of axis-aligned rectangles.

2.2.1. Models of Hook Graphs

We begin this subsection by reminding the reader of what a hook graph is.

Definition 2.2.1. A *hook graph* is a cyclic segment graph which has a generalised hook representation where all the generalised hooks touch the diagonal. We call such generalised hooks in the plane *hooks*.

Remark that hooks can also be horizontal or vertical segments; the only restriction is that hooks must touch the diagonal.

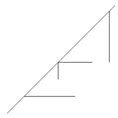


Figure 2.14.: The three different types of hooks.

Remark 2.2.2.

1. *Hook graphs have an PIC representation* $\{(I_1, p_1), (I_2, p_2), \dots, (I_n, n)\}$ *such that for all* i*, we have* $p_i \in I_i$*. This is immediate when obtaining the point-interval-containment representation by projecting the hooks onto the diagonal as described in remark 2.1.14.*
2. *Cyclic segment graphs are hook graphs if and only if they have a cyclic segment representation, whose segments themselves are tangent to the parabola. This is immediate from the proof of Proposition 2.1.7.*

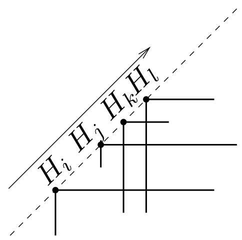


Figure 2.15.: Four hooks with $H_i <_h H_j <_h H_k <_h H_l$.

Given a hook representation $\mathcal{H}$ of a hook graphs G, we call the cyclic segment ordering corresponding to $\mathcal{H}$, a *hook ordering* of G. In poset notation, given a set $\mathcal{H}$ of hooks in the plane, we denote the hook orderings by the poset $(\mathcal{H}, <_h)$. We abuse notation slightly and denote the corresponding ordering of the vertices by $(V, <_h)$. Generally, for a given hook graph the hook ordering is not unique.

We now introduce a completely combinatorial characterisation of hook graphs.

Definition 2.2.3 (Cross Completion Property). Given a graph G, then an ordering of the vertices $v_1, v_2, .., v_n$ is said to satisfy the *cross completion property* if

$$\text{For all } i < j < k < l, \text{ if } (v_i, v_k), (v_j, v_l) \in E \text{ then } (v_j, v_k) \in E. \tag{2.1}$$

Remark 2.2.4. *It is easy to see that an ordering of the vertices satisfies the cross completion property if and only if*

- *for all $i < j < k < l$ such that i (resp. l) is the minimum (resp. maximum) index such that $(v_i, v_k), (v_j, v_l) \in E$ then we must have $(v_j, v_k) \in E$.*

Proposition 2.2.5. *A graph is a hook graph if and only if there exists an ordering of the vertices that satisfies the cross completion property.*

Proof. Given a hook representation of a graph, we show that the hook ordering $v_1, v_2, \ldots, v_n$ satisfies the cross completion property. Assume there exists $i < j < k < l$ such that $(v_i, v_k), (v_j, v_l) \in E$. This implies that H_j must leave the triangle $\triangle$ to the right, where $\triangle$ is the triangle bounded by the diagonal, the vertical part of H_k, and the horizontal part of H_i (see Figure 2.16). By continuity, we must have that H_j must intersect H_k in point p (see Figure 2.16). Therefore $(v_j, v_k) \in E$ and the hook ordering satisfies the cross completion property.

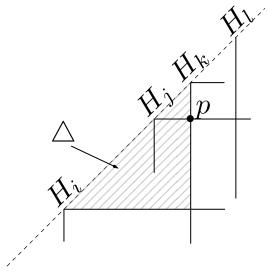


Figure 2.16.: Hooks satisfy the cross completion property.

We now show that an ordering satisfying the cross completion property is a hook ordering of the vertices of the graph. The indexed intervals of the representation are constructed as follows:

$$I_i = [min\{i, \{min\{j : (v_i, v_j) \in E\}\}, \max\{i, \max\{j : (v_i, v_j)\}\}].$$

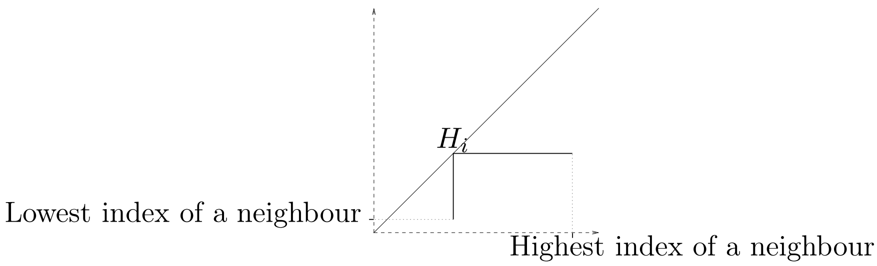


Figure 2.17.: The hook H_i that corresponds to I_i.

We have clearly forced i to be in I_i, so in order to check that these intervals define a hook graph we have to check that

$$(v_j, v_k) \in E \iff j \in I_k \text{ and } k \in I_j \text{ for } j < k.$$

($\Rightarrow$) This is immediate by the definition of the intervals.
($\Leftarrow$) Given the intervals as defined above, assume that there exists $i < j$ such that $i \in I_j$ and $j \in I_i$, but $(v_i, v_j) \notin E$. In this case:

- $i \in I_j \implies$ there exists $k < i$ such that $(v_k, v_j) \in E$, and
- $j \in I_i \implies$ there exists $l > j$ such that $(v_i, v_l) \in E$.

From these two facts, the cross completion property implies that $(v_j, v_k) \in E$, which is a contradiction. Therefore a graph has an ordering with the cross completion property if and only if it is a hook graph. □

Proposition 2.2.6. *Given a graph G, an ordering $<_h$ of the vertices satisfies the cross completion property if and only if we can draw the graph in the plane so that the following hold:*

- *The vertices lie on the x-axis so that the order of the vertices from left to right is $<_h$.*
- *All the edges can be drawn above the x-axis so that if two edges (v, w) and (x, y) intersect with $v < w < x < y$, then (w, x) must be an edge as well (see Figure 2.18).*

We call a drawing as in Proposition 2.2.6, a *hook ordering on a line.*

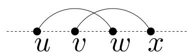


Figure 2.18.: Forbidden ordering if (v, w) is not an edge.

Given an ordering of the vertices V of a graph and four vertices $u <_h v <_h w <_h x$, then we say that the (ordered) 4-tuple, (u, v, w, x), *violates* the cross completion property if (u, w) and (v, x) are both edges, but (v, w) is not an edge. By Proposition 2.2.5, if there is a 4-tuple that violates the cross completion property, then the ordering cannot be a hook ordering of V.

2.2.2. Contained Graphclasses, Forbidden Subgraphs and Basic Properties

We begin this subsection by showing that there are cyclic segment graphs that are not hook graphs.

Proposition 2.2.7. *Hook graphs are a strict subclass of cyclic segment graphs.*

Before we prove this, we prove a lemma about induced disjoint paths in a hook graph. Two paths $x_1, x_2 \ldots x_i$ and $y_1, y_2, \ldots, y_j$ in G are called *induced disjoint paths* if $G|_{\{x_1,\ldots,x_i,y_1,\ldots,y_j\}}$ is the disjoint union of the two paths.

Lemma 2.2.8. *Let $(V, <_h)$ be a hook ordering of a hook graph G and let $x_1, x_2 \ldots x_i$ and $y_1, y_2, \ldots, y_j$ be two* induced disjoint paths. *We have that the two paths cannot alternate in the hook ordering, that is, given 4 distinct numbers $i_1, i_2, j_1, j_2 \in \mathbb{N}$, then the following order of vertices is forbidden:*

- $x_{i_1} <_h y_{j_1} <_h x_{i_2} <_h y_{j_2}$.

Proof. Assume there exist vertices as above, such that $x_{i_1} <_h y_{j_1} <_h x_{i_2} <_h y_{j_2}$. Consider the hook ordering on a line. The edges in the path P from x_{i_1} to x_{i_2} must block the vertex y_{j_1} from y_{j_2} (see Figure 2.19). By this we mean that any path P' in the top half of the plane from y_{j_1} to y_{j_2} would cross an edge in P or go through a vertex in P. However, the paths P and P' do not share a vertex and so there must be an edge in P that intersects an edge in P'. Therefore, by the property of a hook ordering on a line, there must be an edge between a vertex of P and a vertex of P'. This contradicts that the paths P and P' are induced disjoint paths. We conclude that two induced disjoint paths cannot alternate in the hook ordering. □

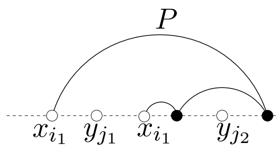


Figure 2.19.: y_{j_1} blocked from y_{j_2} by P.

Using this lemma, we show that the full subdivision of $K_{2,3}$ (see Figure 2.20), denoted $K^{\bullet}_{2,3}$, is not a hook graph. Figure 2.21 shows a cyclic segment representation of $K^{\bullet}_{2,3}$. Once we have shown that $K^{\bullet}_{2,3}$ is not a hook graph we have proved Proposition 2.2.7.

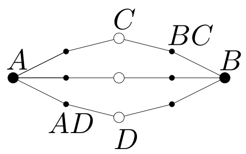


Figure 2.20.: $K^{\bullet}_{2,3}$.

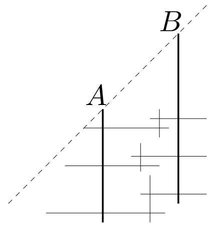


Figure 2.21.: A cyclic segment representation of $K^{\bullet}_{2,3}$.

Proof of Proposition 2.2.7. Suppose that there exists a hook representation of $K^{\bullet}_{2,3}$. Consider the hook ordering cyclically, that is, take the hook ordering on a line then 'glue the two ends of the line together' to obtain the hook ordering on a circle (see Figure 2.22). The vertices labeled A and B naturally partition the other vertices into two sets according to where they lie in this order. By the pidgeonhole principle, two of the three white vertices in Figure 2.20 must lie in the same half. Without loss of generality these are the vertices C and D in Figure 2.20 and they lie on the circle in the order shown in Figure 2.23.

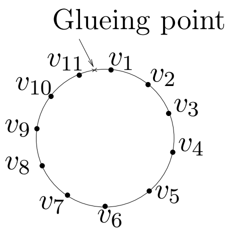


Figure 2.22.: The hook ordering on a circle.

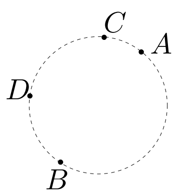


Figure 2.23.: Order of vertices A, B, C, D on the circle.

The paths A, AD, D and B, BD, D are alternating in the hook ordering as we go around the circle. No matter where we break this cyclic order we must still have that these two paths alternate in the hook ordering. However, these two paths are disjoint, which is a contradiction of Lemma 2.2.8. □

Remark 2.2.9. *The proof of Proposition 2.2.7, can be generalised in the following way: Given a graph G and two non-adjacent vertices x and y in G. Then there do not exist three disjoint induced paths P_1, P_2, and P_3 from x to y such that each path has at least 5 vertices, i.e., if $|P_1|, |P_2|, |P_3| \geq 5$ are are paths with no vertices in common, then the edge set of $G|_{P_1 \cup P_2 \cup P_3}$ must contain edges that are not in the paths P_1, P_2, and P_3 (abusing the notation slightly by regarding a path P_i as a set of vertices and as a graph).*

Note that $K_{2,3}^{\bullet}$ is a bipartite graph, therefore by Proposition 2.1.2, we have shown that not all grid intersection graphs are hook graphs. We now show that the graph G shown in Figure 2.25 is not a hook graph. Before we prove that G is not a hook graph, we show that the induced subgraph of G shown in Figure 2.24 has a representation that is fairly restricted.

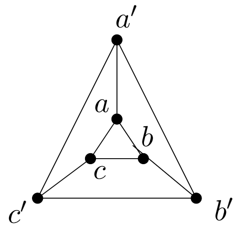


Figure 2.24.: The induced subgraph.

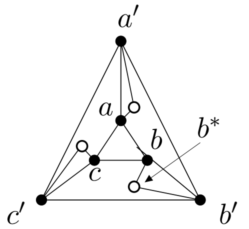
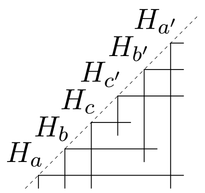
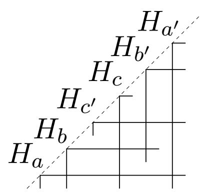


Figure 2.25.: Another non hook graph.

Proposition 2.2.10. *Consider a representation of the graph shown in Figure 2.24 with the vertices named as in the figure. If vertex a is the lowest vertex in the hook ordering and $a <_h b <_h c$, then the only possible canonical hook representations are the ones shown in Figure 2.26.*



Representation 1.



Representation 2.

Figure 2.26.: The two possible canonical hook representations of G.

Proof. Consider the vertices named as in Figure 2.24 and satisfying the conditions of the proposition, then we must have that $c <_h a'$. If this were not true, then either we have $b <_h a' <_h c$, or $a <_h a' <_h b$. In the first case, the (ordered) 4-tuple (a, b, a', c) violates the cross completion property. In the second case, we cannot add c' to the ordering without violating the cross completion property:

1. If $b <_h c'$, then the 4-tuple (a, a', b, c') must violate the cross completion property.

2. If $c' <_h b$, then the 4-tuple (a, c', b, c) must violate the cross completion property.

Note that the second statement above does not depend on the position of a', therefore we must have that $b <_h c'$. We now make the following observations:

- We have that $c <_h b'$. If this were not true, from the fact that $c <_h a'$, the 4-tuple (a, b', c, a') would violate the cross completion property.
- We have that $b' <_h a'$. If this were not true, the 4-tuple (a, b, a', b') would violate the cross completion property. Here we used that $b <_h c <_h a'$ and $b <_h c <_h b'$.
- We have that $c' <_h b'$. If this were not true, the 4-tuple (b, c, b', c') would violate the cross completion property.

Combining the restrictions above, there are only two possible hook orderings given the assumptions on a, b, c. These orderings correspond to the canonical hook representations in Figure 2.26. □

Proposition 2.2.11. *The graph G shown in Figure 2.25 is not a hook graph.*

Proof. Assume that we can find a hook representation of the graph G shown in Figure 2.25. In this case we can find a canonical hook representation of G. Because of the symmetry in the graph, we can assume without loss of generality that the restriction of the hook ordering on the set $\{a, b, c, a', b', c'\}$ satisfies the conditions in the statement of the previous proposition. Therefore, the canonical hook representation restricted to the hooks $\{H_a, H_b, H_c, H_{a'}, H_{b'}, H_{c'}\}$ is one of the two representations in Figure 2.26. Suppose we have representation 1 (the other case is similar), then we cannot place b^* in the ordering without violating the cross completion property:

- If $b^* <_h a$, then the 4-tuple (b^*, a, b', a') violates the cross completion property.
- If $a <_h b^* <_h c$, then the 4-tuple (a, b^*, c, b') violates the cross completion property.
- If $c <_h b^* <_h c'$, then the 4-tuple (c, b^*, c', b') violates the cross completion property.
- If $c' <_h b^* <_h a'$, then the 4-tuple (b, c', b^*, a') violates the cross completion property.
- If $a' <_h b^*$, then the 4-tuple (a, b, a', b^*) violates the cross completion property.

Therefore G is not a hook graph. □

Remark 2.2.12. *If we add edges between any of the white vertices in Figure 2.25, then the new graph that we obtain is also not a hook graph. This is true because we only used the fact that we cannot represent the vertex b^* in the proof that the graph in Figure 2.25 is not hook graph.*

Having mentioned graph classes that are not hook graphs, we now discuss some graph classes that have hook representations.

Proposition 2.2.13. *Outerplanar graphs are hook graphs.*

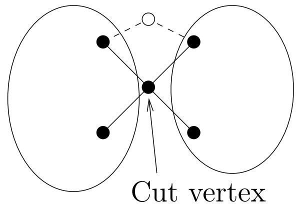


Figure 2.27.: Adding a (white) vertex and edges to remove cut vertices.

Proof. Let G be an outerplanar graph. If there are any cut vertices, we add a vertex to the outer face and connect it to two of the neighbours of the cut vertex that are in different components (see Figure 2.27). We do this in such a way that the graph remains outerplanar by avoiding that we enclose a vertex. This gives us a 2-connected outerplanar graph G' with G as an induced subgraph.

As induced subgraphs of a hook graphs are also hook graphs, it is enough to show that G' is a hook graph. Now, as the graph is 2-connected, the outer face is a cycle. Let $v_1, v_2, \ldots, v_n$ be the order in which we meet the vertices when going around the outer face of an outerplanar embedding of G' in a clockwise direction starting from any vertex v_1. We show that this ordering satisfies the cross completion property.
Given vertices v_i, v_j, v_k, v_l with $i < j < k < l$ then they lie in clockwise order along the outerface. If (v_i, v_k) is an edge, then it must be a Jordan edge going through the bounded region that the outerface defines; if not, one of v_j or v_l would not lie incident to the outerface, which is a contradiction of G' being outerplanar. Similarly, if (v_j, v_l) is an edge, then it must also lie in the bounded region. Therefore, it not possible to have both of these edges in our graph as they would cross in the outerplanar embedding of G'. This proves that the ordering $v_1, v_2, \ldots, v_n$ satisfies the cross completion property. □

Another example of graphs that have hook representations are 2-directional orthogonal ray graphs.

Definition 2.2.14. A 2-*directional orthogonal ray graph (2DORG)* is the intersection graph of horizontal and vertical rays in $\mathbb{R}^2$, where

- A *horizontal ray* is defined as $\{(x, y) : x \leq z\}$ for fixed y, z in $\mathbb{R}$.
- A *vertical ray* is defined as $\{(x, y) : y \geq z\}$ for a fixed x, z in $\mathbb{R}$.

Figure 2.28.: A horizontal ray.

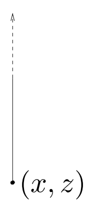


Figure 2.29.: A vertical ray.

2DORGs were introduced in 2008 by Shrestha et al. [55] as a subclass of Orthogonal ray graphs, where an *orthogonal ray graph* is the intersection graph of horizontal and vertical half lines in the plane (note that the rays may go to infinity to the north, east, south, and west). In this paper they give a characterisation of 2DORGs and a characterisation of trees that are 2DORGs. They also show that orthogonal ray graphs are a strict subclass of unit grid intersection graphs, which are defined to be graphs that have a grid intersection representation whose segments are all of the same length. Shrestha et al. [54] published a paper in 2009 explaining some motivation for studying orthogonal ray graphs and used them to model defective nano-wire crossbars. In 2010, Shrestha et al. [56] showed that 2DORGs are exactly bipartite graphs whose complement is a circular arc graph. They then apply a result by Trotter and Moore [43] to show that a graph is a 2-dimensional orthogonal ray graph if and only if its associated bipartite poset has interval dimension at most 2, which implies that 2DORGs have a characterisation using forbidden induced subgraphs (see Trotter [59]). Applying a result by Feder et al. [22] they also show that a graph is a 2DORG if and only if it is chordal bipartite and contains no edge asteroid. Another consequence of this characterisation is that there exists an $O(n^2)$ time algorithm to test if a graph is a 2DORG (from a result by McConnell [42]). For a better understanding of the different characterisations of 2DORGS, see Shrestha et al. [56].

Remark 2.2.15 (2DORGs have hook representations)**.** *If we take a finite* 2*-directional orthogonal ray graph, then we can move the line $f(x) = y$ to the left until we have all the intersections between the rays below it. We can then remove the infinite parts of the rays that all lie above this line to obtain a hook representation of the graph.*

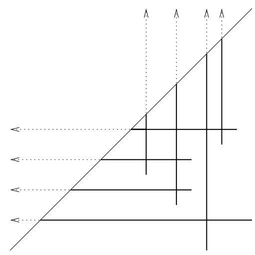


Figure 2.30.: Hook representations of a 2DORG.

We now introduce another subclass of bipartite hook graphs, which we call stick graphs.

Definition 2.2.16. We define a *stick* to be a hook that is a horizontal or a vertical segment. A graph G is a *stick graph* if it is the intersection graph of a set of sticks. Such a hook representation that consists only of sticks is called a *stick representation.* The hook ordering of a stick representation is called a stick ordering. For a vertex v in a stick graph, we denote the stick corresponding to v in a representation by s_v.

Remark 2.2.17. *Above, we have shown that all* 2*DORGs have a special type of stick representation, namely, all the horizontal sticks are before all the vertical sticks in the hook ordering. It is not difficult to see that if all the sticks in a stick representations of a graph G have this additional conditional, then they can be extended to obtain a* 2*DORG representations of G. Therefore, this gives a characterisation of* 2*DORGs.*

As the set of vertical segments and the set of horizontal segments both form an independent set, we get that stick graphs are bipartite. One can show that all cycles with an even number of vertices have stick representations (see Figure 2.31 for a representation of a 6-cycle, denoted C_6). Because 2DORGs do not contain a cycle on more than 4 vertices, it follows that not all stick graphs are 2DORGs. Following a similar argument to the one in Proposition 2.2.5, we can characterise stick graphs as follows.

Proposition 2.2.18. *A graph $G = (V, E)$ is a stick graph if and only if it is bipartite and there exists an ordering $<_s$ of the vertices of G such that:*

1. *The ordering $<_s$ satisfies the cross completion property.*
2. *If A and B are the two colour classes of G, then for all vertices $a \in A$ and $b \in B$ such that (a, b) is an edge of G, we have that $a <_s b$.*

This ordering $<_s$ corresponds to the stick ordering.

Later in Theorem 2.3.28, we describe all the possible hook orderings of a cycle. Applying Theorem 2.3.28, together with the property 2 of the stick ordering stated in Proposition 2.2.18, one can show that the stick ordering of a cycle of even length is unique up to graph automorphisms. We now use the uniqueness of stick orderings of cycles to show that not all bipartite hook graphs have stick representations.

Proposition 2.2.19. *There exists a bipartite hook graph that doesn't have a stick representation.*

Before we show this result, we prove a lemma about the structure of a stick representation of the graph G_6 shown in Figure 2.32, which is the key gadget in the construction of a bipartite hook graph that has no stick representation.

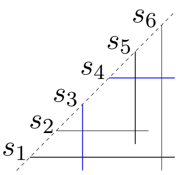


Figure 2.31.: Stick representation of C_6.

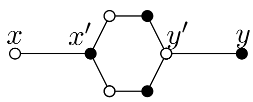


Figure 2.32.: The graph G_6.

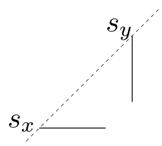


Figure 2.33.: The sticks in case 1.

Lemma 2.2.20. *Given a stick representation of the graph G_6, which is shown in Figure 2.32, if the vertices x and y in Figure 2.32 satisfy $x <_s y$, then one of the two following statements hold:*

1. *The stick s_x is a horizontal segment and the stick s_y is a vertical segment (see Figure 2.33).*
2. *The sticks s_x and s_y are seperated from each other, i.e., any continuous curve in the plane that intersects both s_x and s_y must either* ***cross*** *the diagonal or intersect one of the sticks that is not s_x or s_y in the representation.*

Proof. First note that G_6 is just a 6-cycle with two extra vertices x and y, which are adjacent to two vertices in the 6-cycle. The stick representation of the 6-cycle is unique up to automorphisms of the graph and must be as in Figure 2.31. In G_6, there are two vertices x' and y' that are two opposite vertices in the 6-cycle such that (x, x') and (y, y') are both edges (see Figure 2.32). Therefore, as x' and y' are opposite in the 6-cycle and we supposed that $x <_s y$, the only possible choices in Figure 2.31 for the pair of sticks $(s_{x'}, s_{y'})$ are (s_1, s_5), (s_2, s_6), and (s_3, s_4). If $(s_{x'}, s_{y'}) = (s_3, s_4)$, then s_x is a horizontal segment and s_y is a vertical segment, i.e., the first statement holds. Suppose $(s_{x'}, s_{y'}) = (s_1, s_5)$, then as s_y is a stick that must intersect s_5, it must be completely contained in the interior of the region Ω shown in Figure 2.34. More precisely, Ω is the region that is bounded by s_1, s_3, s_6,

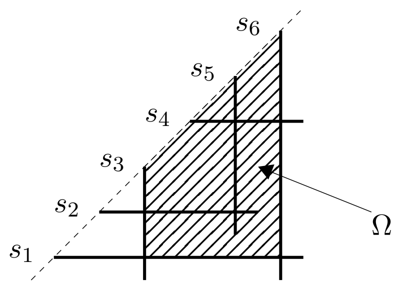


Figure 2.34.: The bounded region Ω.

and the diagonal. As s_x is a stick that intersects s_1, it must be contained in the exterior of Ω. Therefore, any continuous curve that intersects both s_x and s_y must also intersect the boundary of Ω, and the second statement holds. One can show in a similar manner that in the case where $(s_{x'}, s_{y'}) = (s_2, s_6)$, then the second statement must also hold. ☐

Remark 2.2.21. *Consider a stick graph G such that G_6 is an induced subgraph of G and $x <_s y$. If there exists a path P from x to y that does not go through any of the vertices in G_6, then s_x must be horizontal and s_y must be vertical in a stick representation of G. Indeed, the second statement in Lemma 2.2.20 cannot hold because from the stick representation of P we get a continuous curve in the plane that intersects s_x and s_y, but does not cross the diagonal and does not intersect any of the sticks in the representation of G_6.*

Proof of Proposition 2.2.19. We show that the graph G shown in Figure 2.35 does not have a stick representation. A hook representation of G is shown in Figure 2.36. Assume that

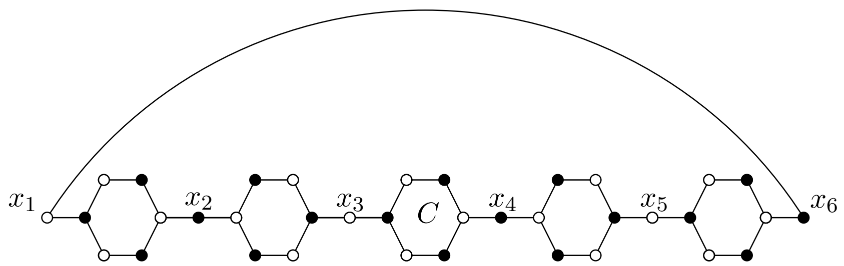


Figure 2.35.: A bipartite hook graph G_{bip} that is not a stick graph.

we can find a stick representation of G. First note that G restricted to the vertices x_3 and

x_4 together with the 6-cycle C (see Figure 2.36) is isomorphic to G_6. Now G also contains a path from x_3 to x_4 that satisfies the conditions of Remark 2.2.21 above. Therefore s_{x_3} and s_{x_4} must be horizontal and vertical sticks respectively. Arguing similarly, but with the cycle C', we can conclude that $x_3 <_s x_2$ and that s_2 is a vertical stick. There are two cases:

1. Case 1: $x_2 <_s x_4$. In which case, we consider the vertex x_1. Again, applying Lemma 2.2.20 and Remark 2.2.21 we get that $x_1 <_s x_2$ and therefore $x_1 <_s x_4$. One can find a path P from x_1 and x_4, and a path P' from x_2 to x_3, so that P and P' are two induced disjoint paths, i.e., none of the vertices in P are adjacent to any of the vertices in P'. By Lemma 2.2.8, we can conclude that P and P' do not alternate in the stick ordering, and therefore we must have $x_1 <_s x_3$. However, if $x_1 <_s x_3$, it is not hard to see that we have two other induced disjoint paths that alternate in the stick ordering. Therefore, Case 1 is not possible.

2. Case 2: $x_4 <_s x_2$. In this case, the same argument as in Case 1 applied to the vertices x_2, x_3, x_4, and x_5 can be used to show that there must be induced disjoint paths that alternate in the stick ordering. Therefore, this case is also not possible.

We can conclude that G is not a stick graph, but is a bipartite hook graph from Figure 2.36. □

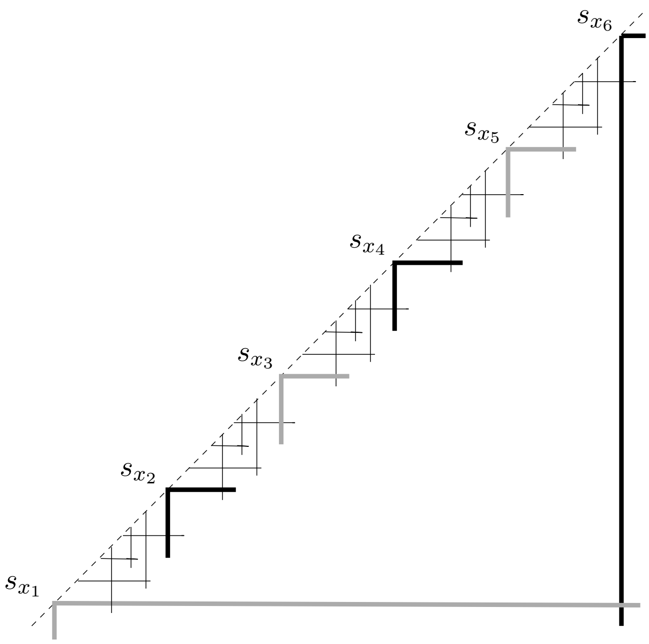


Figure 2.36.: A hook representation of G_{bip}.

Some motivation for the recognition of bipartite hook graphs and stick graphs. In Theorem 2.1.10, we showed that cyclic segment graphs are NP-complete to recognise, but the complexity of the recognition problem for hook graphs remains unknown. The complexity is even unknown for bipartite hook graphs and *stick graphs*. By Remark 2.2.15, we know that 2DORGs are stick graphs and one can recognise them in polynomial time (see

McConnell [42]). Stefan Felsner (personal communication) has proved that a stick graph is the comparability graph of a 3-dimensional posets of height 2. Veit Wiechert (personal communication) has generalised this result by showing that bipartite hook graphs are also comparability graphs of 3-dimensional posets of height 2. Yannakakis [61] showed that testing whether a height 2 poset has dimension at most k is NP-complete for $k \geq 4$. Testing whether a poset is of dimension at most 2 can be done in polynomial time (see Golumbic [27]). It remains unknown, whether testing if a poset of height 2 has dimension at most 3 is NP-complete. This gives much motivation for investigating the complexity of the recognition question for stick graphs and bipartite hook graphs. For further reading about posets and their dimensions, we refer the reader to Trotter [59] and Golumbic [27].

We now show that the class of interval graphs is contained in the class of hook graphs.

Proposition 2.2.22. *Interval graphs have a hook representation.*

As we are considering a finite number of closed intervals, we can peturb an endpoint slightly without changing the intersection graph. Therefore we may assume that all endpoints of intervals are different.

Proof. Place the interval representation of an interval graph G on the x-axis so that all the intervals lie under the diagonal. Lift each interval I vertically to get a horizontal segment S_I whose left endpoint intersects the diagonal. Now add to each S_I, the vertical segment from the left endpoint of S_I to the x-axis to get a hook H_I (see Figure 2.37).

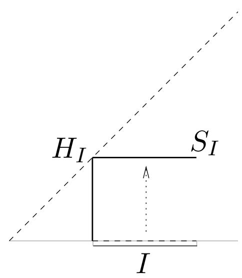


Figure 2.37.: From an interval to a hook.

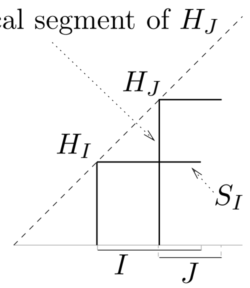


Figure 2.38.: Intersection of H_I and H_J.

Two intervals I and J intersect if and only if the left endpoint of one interval is in the other (w.l.o.g. the left endpoint of J is in I). This is true if and only if the vertical segment of H_J intersects the horizontal segment H_I (see Figure 2.38). Therefore the intersection graph of these hooks is isomorphic to G. Notice, that we do obtain a set of hooks as we may take an interval representation whose left endpoints are all different. □

We conclude this section by showing that hook graphs are a special type of intersection graph of axis-aligned rectangles. An *axis-aligned rectangle* R is a set in $\mathbb{R}^2$ of the form $I_x \times J_y$, where I_x and J_y are intervals in $\mathbb{R}$. Throughout the rest of this chapter, unless stated otherwise, all the rectangles that we mention are axis-aligned, and a *rectangle intersection graph* is an intersection graphs of axis-aligned rectangles. Before we state the characterisation, we discuss some results that are known about rectangle intersection graphs. Rectangle intersection graphs were introduced by Roberts [50] in 1969. The complexity of the recognition problem for rectangle intersection graphs was unknown for a long time, until it was finally proved by Kratochvíl [36] in 1994 that the recognition problem for rectangle intersection graphs is NP-complete. Asano and Imai [3] gave an $O(n \log(n))$ time algorithm to compute the clique number $\omega(G)$ of any rectangle intersection graph G. The independence number $\alpha(G)$ (Fowler et al. [25]), the clique covering number $\gamma(G)$ (Asano and Imai [3]), and the chromatic number $\chi(G)$ (Asano and Imai [3]) have all been shown to be NP-hard to compute for rectangle intersection graphs in general. For a rectangle intersection graph G, the relation between $\chi(G)$ and $\omega(G)$, and the relation between $\gamma(G)$ and $\alpha(G)$ are both very interesting questions, which we discuss further in Subsections 2.3.3 and 2.3.4. In Subsection 2.3.3, we discuss results that approximate the chromatic number by a function of the clique number, which we apply in the case of hook graphs to achieve a bound of $O(\omega(G)\log(\omega(G)))$ on $\chi(G)$. In Subsection 2.3.4, we use the rectangle intersection representation below to discuss approximations of the clique covering number of hook graphs. In particular, we show that for the special class of rectangle intersection graphs given below, the independence number is a 2-approximation of the clique covering number.

Proposition 2.2.23. *A graph is a hook graph if and only if it is the intersection graph of rectangles whose top-left corner lies on the diagonal.*

Proof. The first thing to note is that a rectangle is uniquely defined by its top and left boundaries. One of these boundaries can be a point, in which case we have a horizontal or a vertical segment, which is a degenerate rectangle.
Given a hook, we can obtain a rectangle whose top-left corner is touching the diagonal by letting the vertical (resp. horizontal) part of a hook be the left (resp. top) boundary of the rectangle (see Figure 2.39).

Two hooks. Two rectangles.

Figure 2.39.: From hooks to axis-aligned rectangles touching the diagonal.

We can obtain hooks from rectangles whose top-left corner are touching the diagonal by using the inverse of the map above. Two hooks intersect if and only if the vertical part of one hook intersects the horizontal part of another. In addition, two rectangles whose top-left corner is on the diagonal intersect if and only if the top boundary of one intersects the left boundary of the other. These two facts imply that both the maps above preserve the intersections and therefore the intersection graphs of both sets of objects are the same. ☐

2.3. The Perfect Graph Parameters of Hook Graphs

Given a hook graph G, in this section we investigate properties of the clique covering number γ, the chromatic number χ, the clique number ω, and the independence number α of

G. Before we give an overview of this section, we define a maximum-weight clique and a maximum-weight independent set.

Definition 2.3.1. Given a graph $G = (V, E)$ and a weight function $w : V \to \mathbb{R}$ on the vertices . Let $A \subset V$, then we define the *weight of A* to be

$$w(A) = \sum_{v \in A} w(v).$$

1. A *maximum-weight clique* is a clique C_{max} in G of maximum weight, i.e.,

$$w(C_{max}) = \max\{w(C) : C \text{ is a clique in G }\}.$$

2. A *maximum-weight independent set* is an independent set I_{max} in G of maximum weight, i.e.,

$$w(I_{max}) = \max\{w(I) : I \text{ is an independent set in G }\}.$$

We denote by WMCLIQUE, the problem of computing a maximum-weight clique. Similarly, we denote by WMIS, the problem of computing a maximum-weight independent set.

Note that if we take the weight function to be the map $w(v) = 1$ for all $v \in V$, then maximum-weight cliques and maximum-weight independent sets coincide with maximum cliques and maximum independent sets, respectively.

In Subsection 2.3.1, we give a $O(n^2)$ time algorithm to solve WMCLIQUE for hook graphs. In Subsection 2.3.2 we give an $O(n^3)$ time algorithm to solve WMIS for hook graphs. In Subsection 2.3.3 we discuss approximations of the chromatic number. In particular, we discuss the relation between the chromatic number and the clique number of rectangle intersection graphs and segment graphs. We then apply a result by Chalermsook [10] on rectangle intersection graphs to give an $\log(\omega(G))$-approximation algorithm of the chromatic number. We also include a sketch proof of the result by Chalermsook [10] that we use. In Subsection 2.3.4, we show that the independence number is a 2-approximation of the clique covering number. All of the algorithms given in this section are polynomial time algorithms. Unless stated otherwise, the time complexity of the algorithms in this chapter are assumed to be with respect to the number of vertices in the input.

2.3.1. A Polynomial Time Algorithm for Weighted Maximum Clique

In this subsection we give an $O(n^2)$ time algorithm that solves WMCLIQUE for hook graphs given a hook ordering as input. There already exists an $O(n \log(n))$ time algorithm to find a maximum clique in hook graphs from the $O(n \log(n))$ time algorithm that is given by T. Asano and H. Imai [3]. The algorithm we give uses a different and solves the general case of finding a maximum-weight clique. In our algorithm, we use a nice interval graph structure of the upper-neighbourhood of a vertex (defined below).

Definition 2.3.2. Given a hook graph $G = (V, E)$ with hook ordering $<_h$, and a vertex $v \in V$:

- $U(v) = \{w \in V : (w, v) \in E \text{ and } v <_h w\}$ is called the *upper-neighbourhood* of v.

The hooks in $U(v)$ (in black). The hooks in $L(v)$ (in black).

Figure 2.40.: The upper and lower-neighbourhood of a vertex.

- $L(v) = \{w \in V : (w, v) \in E \text{ and } w <_h v\}$ is called the *lower-neighbourhood* of v.

Lemma 2.3.3. *Given the ordering $(V, <_h)$ of the vertices of a hook graph, then for all $v \in V$:*

1. *$G|_{U(v)}$ is an interval graph.*
2. *$G|_{L(v)}$ is an interval graph.*
3. *Given $w \in V$ with $v <_h w$, we have that $G|_{U(v) \bigcap L(w)}$ is a clique.*

Proof of (1). For $x \in U(v)$, project the horizontal part of the hook H_x downwards onto the x-axis to get an interval I_x. Loosely speaking, this is the inverse of the map we used to obtain hooks from an interval representation. Let H_x and H_y be hooks of two vertices x and y in the upper-neighbourhood of a vertex v with $x <_h y$. $H_x \bigcap H_y \neq \emptyset$ if and only the horizontal part of H_x lies beneath the horizontal part of H_y (as they both intersect H_v, which lies beneath both H_x and H_y). However, this is true if and only if the projected horizontal parts intersect each other.
Proof of (2.) For $x \in L(v)$ we project the vertical parts of the hooks onto the y-axis. This gives us an interval representation of $L(v)$ by arguing similarly to 1.
Proof of (3.) Fix $x, y \in U(v) \bigcap L(w)$ with $x <_h y$. As we have $v <_h x <_h y <_h w$ and $(v, y), (x, w) \in E$, the cross completion property implies that (x, y) is also an edge in G therefore $G|_{U(v) \bigcap L(w)}$ is a clique. □

A slightly stronger statement also can be shown with exactly the same arguments: Given a hook ordering $<_h$ of a hook graph G, we have that $G|_{\overline{U}(v)}$ and $G|_{\overline{L}(v)}$ are both interval graphs, where $\overline{U}(v)$ and $\overline{L}(v)$ are defined below.

- $\overline{U}(v)$ is the set of vertices w with $v <_h w$ such that w has a neighbour w' with $w' <_h v$.
- $\overline{L}(v)$ is the set of vertices w with $w <_h v$ such that w has a neighbour w' with $v <_h w'$.

In addition, given $v <_h w$ then one can show that $G|_{\overline{U}(v) \bigcap \overline{L}(w)}$ is a clique.

Weighted Maximum Clique for interval graphs We now show an $O(n)$ time algorithm that finds the maximum-weight clique of an interval graph. For this algorithm we need an interval representation that has already been sorted in a certain way.

Lemma 2.3.4. *Let $w : V \to \mathbb{R}$ be a weight function on the vertex set of an interval graph G. Given the list of endpoints of an interval representation of G in increasing order (with respect to the natural ordering of the real numbers), Algorithm 2 solves WMCLIQUE for G in $O(n)$ time. Here, a list of endpoints contains the additional information of whether an endpoint it is a left or a right endpoint of an interval.*

Algorithm 2 IntWeightedMaxClique

Input: List of endpoints of the intervals in a representation of graph G in increasing order $x_1, x_2, \dots, x_{2n}$

Output: The size of a maximum clique in G

$CMAX \leftarrow \emptyset$
$MAX \leftarrow 0$
$CWEIGHT \leftarrow 0$
$C \leftarrow \emptyset$
for $j = 1, 2, \dots 2n$ **do**
  $v \leftarrow$ vertex whose interval has endpoint x_j
  if Endpoint x_j is a left endpoint **then**
    $CWEIGHT \leftarrow CWEIGHT + w(v)$
    $C \leftarrow C \bigcup \{v\}$
  else
    $CWEIGHT \leftarrow CWEIGHT - w(v)$
    $C \leftarrow C \setminus \{v\}$
  if $CWEIGHT > MAX$ **then**
    $MAX \leftarrow CWEIGHT$
    $CMAX \leftarrow C$
return $CMAX$

Proof. Notice that if we have a clique in an interval graph then the left endpoint of an interval that is furthest to the right is contained in all of the intervals involved in the clique. Figure 2.41 shows four intervals which form a clique. Note that in this example, p is the left endpoint which is contained in all other intervals of the clique.

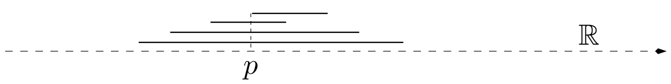


Figure 2.41.: The left endpoint p that is contained in all intervals of the clique.

We define a counter MAX, which is initialised to zero at the beginning of the algorithm. We then go through the list of endpoints in increasing order, such that when we encounter a left endpoint we look at the sum of the weights of the intervals that the endpoint is contained in. We achieve this by keeping another counter $CWEIGHT$, which is increased by the weight of the interval whose left endpoint we meet, or is decreased by the weight of the interval whose right endpoint we meet (line 8 and line 11). That is, $CWEIGHT$ corresponds to the sum of the weights of the intervals whose left endpoint we have visited and whose right endpoint we have not visited. The maximum-weight clique is found similarly and stored in the variable $CMAX$, and its weight is stored in the variable MAX (line 14). Looking at each endpoint and processing it takes a constant amount of time, and as there are $2n$ endpoints, we conclude that the complexity of the algorithm is $O(n)$. ☐

Now we are ready to state the algorithm HookWeightedMaxClique, which solves WMCLIQUE for hook graphs in $O(n^2)$ time given a hook representation as input.

Algorithm 3 HookWeightedMaxClique

Input: List of vertices $v_1, v_2, .., v_n$ corresponding to a hook ordering of a hook graph G together with a weight function $w : V \to \mathbb{R}$

Output: The size of a maximum clique in G

1: $MAX \leftarrow 0$
2: $MAXCLIQUE \leftarrow \emptyset$
3: Initiate the upper-neighbourhoods of all vertices v_j and save them as $U(v_j)$
4: **for** $j = 1, 2, \ldots n$ **do**
5:     $JMAX \leftarrow w(IntWeightedMaxClique(U(v_j))) + w(v_j)$
6:     $JMAXCLIQUE \leftarrow IntWeightedMaxClique(U(v_j)) \bigcup \{v_j\}$
7:     **if** $JMAX > MAX$ **then**
8:         $MAX \leftarrow JMAX$
9:         $MAXCLIQUE \leftarrow JMAXCLIQUE$
10: **return** $MAXCLIQUE$

Theorem 2.3.5. *Given a hook ordering $v_1, v_2, \ldots, v_n$ of a hook graph G and a weight function $w : V \to \mathbb{R}$, the algorithm HookWeightedMaxClique solves WMCLIQUE for G in $O(n^2)$ time.*

Proof. The idea is that we run through the vertices in increasing order with respect to $<_h$, finding the largest clique in their upper-neighbourhoods. A maximum-weight clique must be encountered when we look at the upper-neighbourhood of its lowest vertex in the hook ordering. Therefore if we find a maximum-weight clique in the upper-neighbourhoods of each vertex, then a maximum-weight clique of the entire graph will have been found. By Lemma 2.3.3, the upper-neighbourhood of each vertex is an interval graph. We would like to use IntWeightedMaxClique, but first we need to find and store the upper-neighbourhoods in the correct format for the algorithm, which is done in Line 3. More precisely, we start by associating to each vertex v_j, the inteval $I_j = [j \ , \max\{j \ , \max\{k : (v_j, v_k) \in E\}\}]$. This corresponds to the interval representation that can be found in Lemma 2.3.3 from the representation found as in Figure 2.17. This can be done in $O(n^2)$ time. Then order all the endpoints with respect to the natural order on the real numbers, which can be done in $O(n \log(n))$ time using a basic sorting algorithm. We then find the upper-neighbourhood of each hook, which takes $O(n^2)$ time for all the hooks altogether. This is because checking adjacencies and comparing hook indices can be done in constant time. Once we have all of the upper-neighbourhoods, we go through the ordered list of endpoints above and select the endpoints corresponding to the hooks in the upper-neighbourhood in $O(n^2)$ time, retaining the order of the endpoints. The entire process described so far is what we refer to in Line 3 of HookWeightedMaxClique. Line 5 computes the largest weight of a clique that has v as its lowest vertex in the hook ordering. We go through all of the hooks, keeping track of the largest clique encountered so far (Line 9). As mentioned at the start, the maximum-weight clique will be found when looking at the upper neighbourhood of its lowest vertex v. From Algorithm 2, the maximum-weight clique in an upper-neighbourhood can be found in linear time. As this is computed n times, the FOR LOOP in the algorithm is completed in $O(n^2)$ time. Hence, the entire algorithm can be performed in $O(n^2)$ time. □

2.3.2. A Polynomial Time Algorithm for Weighted Maximum Independent Set

Given a hook representation of a hook graph $G = (V, E)$, together with a weight function $w : V \to \mathbb{R}$, we give an $O(n^3)$ time algorithm that finds a maximum-weight independent set. The key observation that we use is Lemma 2.3.7, which allows us to break down larger instances of the problem into smaller instances and hence use a dynamic programming approach to solve WMIS for hook graphs. We must first define free hooks.

Definition 2.3.6. Given a hook graph $G = (V, E)$ and a hook ordering $v_1, v_2, \ldots, v_n$, let $I_1, I_2, \ldots, I_n$ be the canonical indexed interval representation of G (see Definition 2.1.8) and $H_1, H_2, \ldots, H_n$ be the canonical hook representation of G. Let $\mathcal{I}$ be an independent set in G and let v_i be a vertex in $\mathcal{I}$ with H_i and I_i its hook and indexed interval, respectively.

- We say that H_i is *free from below (in $\mathcal{I}$)* if we can extend the vertical part of the hook downwards to infinity without intersecting any other hook in the representation restricted to the independent set (see Figure 2.42). Equivalently, there is no vertex $v_j \in \mathcal{I}$ with $j < i$ whose indexed interval I_j contains i.
- We say that H_i is *free from the right (in $\mathcal{I}$)* if we can extend the horizontal part of the hook to infinity to the right without intersecting any other hook in the representation restricted to the independent set (see Figure 2.43). Equivalently, there is no vertex $v_j \in \mathcal{I}$ with $i < j$ whose indexed interval I_j contains i.
- We say that H is *free (in $\mathcal{I}$)* if it is free from below and it is free from the right (see Figure 2.44). Equivalently, there is no vertex $v_j \in \mathcal{I}$ whose indexed interval I_j contains i.

Lemma 2.3.7. *Given a hook graph G with hook ordering $v_1, v_2, \ldots, v_n$, every independent set I must have a* free *hook.*

Proof. Consider the restriction of the representation to the independent set $\mathcal{I}$. It is enough to prove the following statement:

- Given a hook H_i that is free from below, then either it is free or there exists another hook H_j with $H_i <_h H_j$ that is free from below.

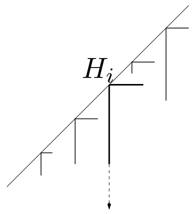


Figure 2.42.: Free from below.

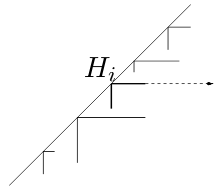


Figure 2.43.: Free from the right.

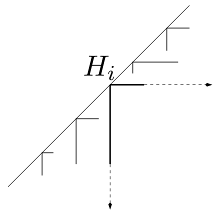


Figure 2.44.: A free hook.

This is sufficient because H_1 is free from below. Therefore, by repeatedly using the statement above we must obtain a free hook as G is finite (see Figure 2.45).

Let H_i be a hook that is free from below. If H_i is not free from the right, there is a vertex $v_j \in \mathcal{I}$ with $i < j$ whose indexed interval I_j contains i (see Figure 2.45). We claim that H_j is free from below. Suppose not, then there exists a vertex $v_k \in \mathcal{I}$ with $k < j$ and $j \in I_k$. As $(k, j) \notin E$, we must have $k \notin I_j$ and therefore we must have $k < i$ as $[i, j] \subset I_j$. However, this implies that $i \in I_k$ because $i \in [k, j] \subset I_k$. This contradicts that v_i is free from below. $\square$

Theorem 2.3.8. *Given a weight function $w : V \to \mathbb{R}$ and a hook ordering of a hook graph $v_1, v_2, \ldots, v_n$, we can find a maximum-weight independent set in $O(n^3)$ time.*

As mentioned above, our algorithm will reduce larger instances into smaller instances of the problem. We will use open triangles, which are defined as follows.

Definition 2.3.9. Let $G = (V, E)$ be a hook graph with a hook ordering $v_1, v_2, \ldots, v_n$. Let $H_1, H_2, \ldots, H_n$ be the canonical hook representation of G. We define the open triangle $\triangle_{i,k}$ to be the interior of the region in the plane enclosed by the triangle whose corners are (i, i), (k, k), and (k, i) (see Figure 2.47). We define $V_{i,k} \subset V$ to be the set of vertices v_j whose hooks H_j lie completely within the region $\triangle_{i,k}$.
We define $\triangle_{0,k}$ and $\triangle_{k,n+1}$ to be the set of hooks which lie to the left of the vertical line $\{(x, k) : x \in \mathbb{R}\}$ and to the right of the horizontal line $\{(k, y) : y \in \mathbb{R}\}$, respectively. We also define $\triangle_{0,n+1}$ to be the vertex set V.

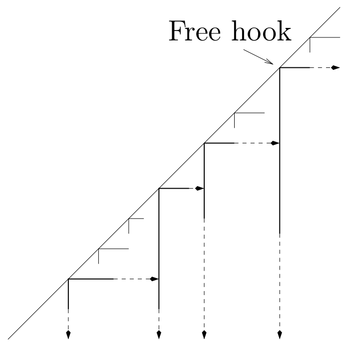


Figure 2.45.: Finding a free hook.

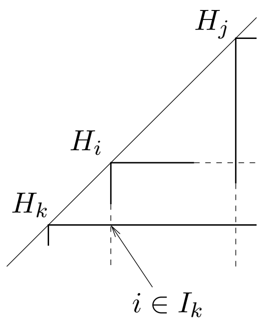


Figure 2.46.: If H' is not free from below.

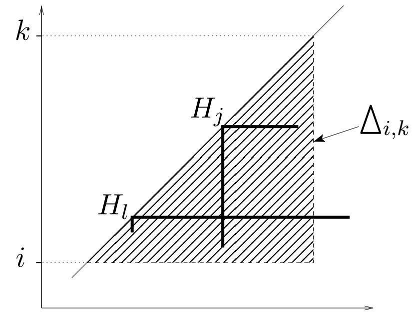


Figure 2.47.: The open triangle $\triangle_{i,k}$ with $H_j \subset \triangle_{i,k}$ and $H_l \not\subseteq \triangle_{i,k}$.

If $I_1, I_2, \ldots, I_n$ is an indexed interval representation of a hook graph, we have that $H_j \subset \triangle_{i,k}$ if and only if $I_j \subset]i,k[$, where $]i,k[$ denotes the open interval between i and k.

Given a canonical hook representation, then $V_{i,k}$ can also be defined as follows: $v_j \in V_{i,k}$ if and only if

- $i < j < k$ and
- $(v_j, v_l) \in E \implies i < l < k$.

Define $w(i,k)$ to be the weight of a maximum-weight independent set of $G|_{V_{i,k}}$, i.e., the largest weight of an independent set whose hooks lie completely in the open triangle $\triangle_{i,k}$.

Remark 2.3.10. $w(i,k) = MAX\{w(j) + w(i,j) + w(j,k) : i < j < k, h_j \subset (i,k)\}$.

Proof. By Lemma 2.3.7, we get that any independent set has a free hook, in particular, there is a free hook H_j in a maximum-weight independent set $\mathcal{I}_{i,k}$ in $G|_{V_{i,k}}$. In this case, by definition of a free hook, none of the other indexed intervals I_l of the vertices $v_l \in \mathcal{I}_{i,k}$ contain j and therefore v_l must be in $V_{i,j}$ or $V_{j,k}$. Now as $\mathcal{I}_{i,k} \bigcap V_{i,j}$ and $\mathcal{I}_{i,k} \bigcap V_{j,k}$ are an independent set in $V_{i,j}$ and $V_{j,k}$, respectively, we must have

$$w(i,k) \leq MAX\{w(j) + w(i,j) + w(j,k) : i < j < k, h_j \subset (i,k)\}.$$

As an independent set in $V_{j,k}$ and an independent set in $V_{i,j}$ together with the vertex v_j is an independent set in $V_{i,k}$, we also have that

$$w(i,k) \geq MAX\{w(j) + w(i,j) + w(j,k) : i < j < k, h_j \subset (i,k)\}.$$

Therefore we have equality and we are done. ☐

Now we are ready to understand the cubic time algorithm that finds a maximum-weight independent set of hook graphs.

Proof of Theorem 2.3.8. We claim that the following algorithm is a cubic time algorithm whose output is a maximum-weight independent set of a hook graph G.

Algorithm 4 HookWeightedMaxIndependentSet

Input: List of vertices $v_1, v_2, .., v_n$ corresponding to a hook ordering of a hook graph G and a weight function $w : V \to \mathbb{R}$

Output: A maximum-weight independent set of G

Compute the canonical indexed interval representation $I_1, I_2, \ldots, I_n$ of the hook ordering
for $k = 1, 2, \ldots n+1$ **do**
    for $i = 0, 1, \ldots, n+1-k$ **do**
        $w(i, i+k) \leftarrow 0$
        $\mathcal{I}_{i,i+k} \leftarrow \emptyset$
        for $j = i+1, i+2, \ldots, i+k-1$ **do**
            if $w(i, i+k) < w(i, i+j) + w(v_{i+j}) + w(i+j+1, i+k)$ AND $I_{i+j} \subset]i,k[$ **then**
                $w(i, i+k) \leftarrow w(i, i+j) + w(v_{i+j}) + w(i+j+1, i+k)$
                $\mathcal{I}_{i,i+k} \leftarrow \mathcal{I}_{i,i+j} \bigcup \{v_{i+j}\} \bigcup \mathcal{I}_{i+j+1,i+k}$
return $\mathcal{I}_{0,n+1}$

Line 1 is carried out once and computes the canonical indexed interval representation in quadratic time. For a fixed i and k, the FOR LOOP in Line 6 calculates $w(i, i+k)$ in linear time as there are k iterations and each iteration can be performed in constant time. The iterations take constant time as they use stored values of $w(i, i+j)$ and $w(i+j+1, i+k)$. In line 7 we check whether $I_j \subset]i, k[$, i.e., we check whether v_j is in $V_{i,k}$. A maximum-weight independent set $\mathcal{I}_{i,i+k}$ in $V_{i,k}$ is stored in line 9. The value $w(i, i+k)$ is calculated for all values of i and k such that $0 \leq i < k \leq n+1$. Therefore HookWeightedMaxIndependentSet is a cubic time algorithm. Note that we initialise the values of $w(i, i+k)$ in line 4. Also note that for all i, one should also initialise $\mathcal{I}_{i,i}$ and $w(i,i)$ to be the emptyset and zero respectively, as these values are sometimes used in the algorithm.
This algorithm computes the maximum-weight independent set of G due to Remark 2.3.10, together with the fact that $V_{0,n+1} = V$ and hence $\mathcal{I}_{0,n+1}$ is a maximum-weight independent set, which is the output. $\square$

Note that we store each value of $w(i, i+k)$ in an array to obtain a faster algorithm by avoiding repetition of many calculations, i.e., we use dynamic programming. However, this could be a problem because we might use a large amount of memory space to store this information in an array.

2.3.3. Approximating the Chromatic Number

In this subsection we explore the relation between the chromatic number χ and the clique number ω of hook graphs. More precisely, we search for a function f such that for all hook graphs G, we have $\chi(G) \leq f(\omega(G))$. For graphs in general, Mycielski [44] showed that such a function f does not exist. However, hook graphs have nice features that one can exploit to show that $\chi(G) = O(\omega(G)\log(\omega(G)))$. This bound comes from an application of a result by Chalermsook [10] about the chromatic number of rectangle intersection graphs. It has been shown by Cantazaro et al. [9] that computing the chromatic number of a hook graph is NP-hard. A consequence of the bound in this subsection is that the clique number is a $O(\log(\omega(G)))$-approximation of the chromatic number as $\omega(G)$ is a lower bound of $\chi(G)$, and cliques have hook representations. The proof of this bound also gives a polynomial time algorithm that computes a $O(\log(\omega(G)))$-colouring of hook graphs. In the case of segment intersection graphs it was a long standing open problem, which was posed by Erdős in the 1970s, whether such a function exists. Recently, the question was answered negatively by Pawlik et al. [49]. They give a construction of a triangle-free segment graph G_k whose chromatic number is k. We will begin this subsection by explaining their construction. First, we need the following definition.

Definition 2.3.11. Given real number a, b, and c, we define a *probe* P to be a subset of $\mathbb{R}^2$ of the form $P = \{(x, y) : x \leq a,\ b \leq y \leq c\}$.

Remark 2.3.12. *Probes are not used here to correspond to vertices, but to define some structure in the representations so that we can recursively construct* G_k.

Theorem 2.3.13. *For each natural number k there exists a triangle-free segment graph G_k with $\chi(G_k) = k$, for which we can find a segment representation together with a set of disjoint probes that satisfy the following properties:*

1. *For a given probe, the set of segments it intersects is an independent set.*
2. *Given any good colouring of G, one of the probes must intersect a set of segments that have been coloured using at least k different colours.*
3. *If a segment intersects a probe, then it must intersect the top and the bottom boundary of the probe.*

Proof. We prove this by induction on k. For $k = 1$, the result is clear as we can just take a probe and a vertical segment that crosses the probe.

$\mathbf{G_{k+1}}$ from $\mathbf{G_k}$: Given the graph G_k with the properties above, we construct the graph G_{k+1} in the following way:

- Take a copy, $\mathcal{S}_k$, of the segment representation of G_k that has the properties of the induction hypothesis.
- For each probe P that comes with $\mathcal{S}_k$, insert a scaled copy of $\mathcal{S}_k$, call it $\mathcal{S}_k^P$, so that it fits in the axis-aligned rectangle R_P at the end P (see Figure 2.48). By scaling $\mathcal{S}^P$, also ensure that the top and bottom boundaries of the probes defined with $\mathcal{S}^P$ intersect the right boundary of P (see Figure 2.49).
- For each probe P' in each small copy of $\mathcal{S}_k$ we add the segment s'_P (see Figure 2.50), which is defined below.

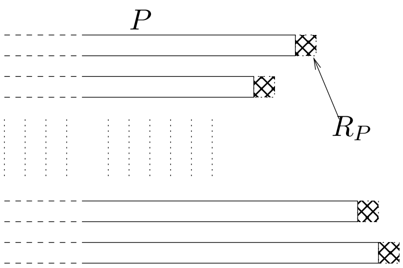


Figure 2.48.: The big picture.

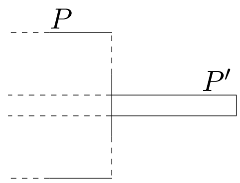


Figure 2.49.: Big and small probe intersection.

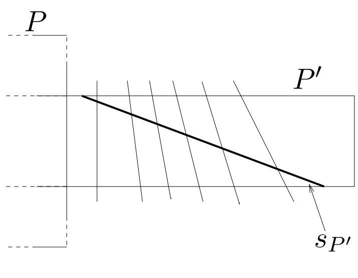


Figure 2.50.: The segment $s_{P'}$.

Defining $\mathbf{s'_P}$: Let P be a probe of $\mathcal{S}_k$ and P' be a probe of $\mathcal{S}_k^P$. Let $I_{P'}$ be the set of segments in $\mathcal{S}_k^P$ that intersect P'. By the induction hypothesis, $I_{P'}$ corresponds to an

independent set. We define one endpoint of the segment $s_{P'}$ to be on the top boundary of P' to the right of $P \bigcap P'$ and to the left of all the intersections points between segments in $I_{P'}$ and P'. The other endpoint of $s_{P'}$ is defined to lie on the bottom boundary of P' to the right of the intersection point $P' \bigcap s$, for all $s \in I_{P'}$.
For G_{k+1}, we have that $\chi(G_{k+1}) \geq k+1$. Indeed, given a good colouring of G_{k+1}, then it induces a good colouring of $\mathcal{S}_k^P$. By the induction hypothesis, there is a probe P' intersecting a set of segments in $\mathcal{S}_k^P$ that have been coloured by at least k different colours. Now as the segment $s_{P'}$ intersects all these segments, $s_{P'}$ must receive a new colour. $\chi(G_{k+1}) \leq k+1$ as we can colour all the segments in every copy of $\mathcal{S}_k$ with the same k colours as they are all disjoint, then use one extra colour to colour the $s_{P'}$, which form an independent set as the probes are all disjoint. G_{k+1} is triangle-free as each copy of $\mathcal{S}_k$ is disjoint, and all the extra segments $s_{P'}$ intersect an independent set by property 1 in the induction hypothesis.

Defining the probes of $\mathbf{G_{k+1}}$: For each probe P' in each $\mathcal{S}_k^P$, we define two probes, P'_{upper} and P'_{lower}, which are both subsets of P'. More precisely, we take P'_{upper} to be a probe lying in the upper part of P' that is crossed by $s_{P'}$, but does not intersect any other segments in the independent set $I_{P'}$ in $\mathcal{S}_k^P$ (see Figure 2.51). Define P'_{lower} to be a probe lying in the lower part of P' that intersects every segment that P' intersects except $s_{P'}$ (see Figure 2.51). The set of probes of G_{k+1} is defined to be the set of all P'_{upper} and P'_{lower}. We show below that the probes satisfy the induction hypothesis.

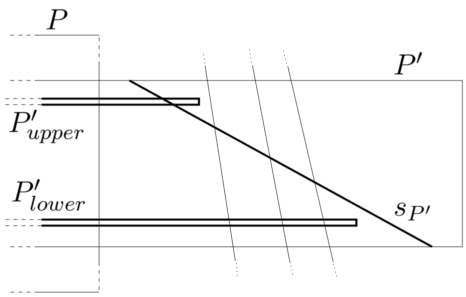


Figure 2.51.: P'_{upper} and P'_{lower}.

Proof of Property 1*:* P'_{upper} intersects $s_{P'}$ and the independent set of segments that P intersects in $\mathcal{S}_k$. Therefore P'_{upper} intersects an independent set. P'_{lower} intersects an independent set as it intersects an independent set in $\mathcal{S}_k^P$ and an independent set in $\mathcal{S}_k$, which are disjoint.
Proof of Property 2*:* Given a good colouring of G_{k+1}, there must exist a probe P of $\mathcal{S}_k$ that intersects a set I_P of segments with k different colours by property 2 of the induction hypothesis applied to $\mathcal{S}_k$. Similarly, there must be a probe P' of $\mathcal{S}_k^P$ that intersects a set $I_{P'}$ of segments in $\mathcal{S}_k^P$ with k different colours. If the sets of colours used on $I_{P'}$ and I_P are not the same, then the set of segments that P_{lower} intersects has been coloured using at least $k+1$ different colours. If the sets of colours used on $I_{P'}$ and I_P are the same, then set of segments that P_{upper} intersects has been coloured using at least $k+1$ different colours because the colour of $s_{P'}$ is different to the colour of all the segments in $I_{P'}$.
Proof of Property 3*:* Let P'_{upper} be a probe of G_{k+1} that intersects probe P of $\mathcal{S}_P^k$. Using the

notation in the proof of Property 2, we have that P_{upper} only intersect $I_P \bigcup \{s_{P'}\}$. Applying Property 3 of the induction hypothesis to $\mathcal{S}_k$, the segments in I_P must cross the top and bottom of P, and therefore the top and bottom of P_{upper}. In addition, we can choose P'_{upper} so that $s_{P'}$ crosses it correctly. Similarly, the segments in I_P cross P_{lower} correctly. The segments in $I_{P'}$ also cross P_{lower} correctly, because of property 2 of the induction hypothesis applied to $\mathcal{S}_k^P$. As P_{lower} only intersects $I_P \bigcap I_{P'}$, we have shown Property 3. This completes the proof. $\square$

Remark 2.3.14. *Although the construction is not very difficult to understand, the difficulty in proving this theorem is finding the correct induction hypothesis using the probes.*

The chromatic number of intersection graphs of axis-aligned rectangles

Hook graphs are special types of segment graphs and it is not obvious whether the construction shown above is realisable using hooks. In fact, it is not possible because hook graphs have rectangle intersection representations (see Proposition 2.2.23) and Asplund and Grünbaum [2] have shown that $\chi(G) \leq 4(\omega(G))^2 - 3\omega(G)$. We will discuss this result later, but first we mention a nice nice property of representations of cliques in rectangle intersection graphs, which comes as a consequence of this Helly's Theorem.

Theorem 2.3.15 (Helly's Theorem [31]). *Let $n \geq d+1$ and $C_1, C_2, \ldots, C_n$ be convex regions in $\mathbb{R}^d$. If every $d+1$ of these sets has a common intersection point, then $\bigcap_{i \leq n} C_i \neq \emptyset$.*

Note that rectangles $R_i = I_i \times J_i$ and $R_j = I_j \times J_j$ intersect if and only if $I_i \bigcap I_j \neq \emptyset$ and $J_i \bigcap J_j \neq \emptyset$. Using this, we can conclude the following remark.

Remark 2.3.16. *Let $R_1, R_2, \ldots, R_n$ be axis-aligned rectangles which correspond to a clique $\mathcal{C}$ in a rectangle intersection graph, then $\bigcap_{i \leq n} R_i \neq \emptyset$.*

Proof. Let $R_i = I_i \times J_i$. As $\mathcal{C}$ is a clique, we have that $R_i \bigcap R_j \neq \emptyset$ for all $i \neq j$. This implies that $I_i \bigcap I_j \neq \emptyset$ for all $i \neq j$. Because an interval is a convex set in $\mathbb{R}$, Helly's Theorem implies that they must have a common intersection point x. Similarly the J_i have a common intersection point y. By definition the point (x, y) intersects all the rectangles in the clique and we are done. $\square$

This is a nice geometric property that one can exploit when dealing with cliques in rectangle intersection graphs. Chalermsook [10] makes use of this property in a result that we apply to hook graphs later in this subsection. First we discuss some notions that help us bound the chromatic number of rectangle intersection graphs. Recall that we can bound the chromatic number of a graph by a number k, by showing that it is $(k-1)$-degenerate (see Remark 1.3.2). Given a rectangle representation of a graph G, one can show that G is $O(w(G))$-degenerate by showing that the representation is s-sparse for some number s. This concept was introduced in [10] and is a generalisation of an idea used in Asplund and Grünbaum [2], where they show $\chi(G) = O((\omega(G))^2)$ for rectangle intersection graphs.

Definition 2.3.17. Given a set $\mathcal{R}$ of rectangles in the plane and a natural number s, then $\mathcal{R}$ is *s-sparse* if for all rectangles R in $\mathcal{R}$ we can assign a set of points $\{P_1^R, P_2^R, \ldots, P_s^R\}$ in the plane such that

- $R, R' \in \mathcal{R}$ with $R \bigcap R' \neq \emptyset \implies \exists i$ such that $P_i^R \in R'$ or $P_i^{R'} \in R$.

This set of points is called a set of *representative points* for a rectangle.

Lemma 2.3.18 (Chalermsook [10]). *Let $\mathcal{R}$ be a set of rectangles in the plane. If $\mathcal{R}$ is s-sparse, then the intersection graph $G = (V, E)$ of $\mathcal{R}$ is $2s\omega(G)$-degenerate, and therefore:*

$$\chi(G) \leq 2s\omega(G) + 1.$$

Proof. The proof of this uses a double counting argument very similar to one used by Asplund and Grünbaum [2]. Namely, we double count the number of elements in the set X, which is defined as follows:

$$X = \{(P^R, e) : e \in E, e = (R, R') \text{ and } P_i^R \in R'\}.$$

Note that we abuse the notation slightly and say that an edge e equals (R, R') if $e = (v_R, v_{R'})$, where v_R and $v_{R'}$ are the vertices that correspond to the rectangles R and R' respectively.

1. By the definition of sparse, for each edge $e \in E$ there exists at least one point P_i^R such that $P_i^R \in R'$, where $e = (R, R')$. Therefore we have $|X| \geq |E|$.
2. Each rectangle has been assigned s points, and there are $|V|$ rectangles altogether, therefore the total number of points is $s|V|$. Each point P_i^R can be in at most $\omega(G)$ different rectangles as the set of rectangles that contain P_i^R form a clique. This implies that each point P_i^R can be in at most $\omega(G)$ pairs in X. Summing this bound over all the points we get $|X| \leq s|V|\omega(G)$.

Putting 1 and 2 together, we get

$$|E| \leq s|V|\omega(G) \implies \frac{|E|}{|V|} \leq s\omega(G). \tag{2.2}$$

Let δ_{ave} denote the average degree of a vertex. Using Equation 2.2 together with the Degree Sum Formula (Proposition 1.1.7) we get

$$\delta_{ave} = \frac{\sum_{v \in V} \deg(v)}{|V|} = \frac{2|E|}{|V|} \leq 2s\omega(G). \tag{2.3}$$

Therefore, the average degree is bounded by $2s\omega$, which means there is a vertex v with $\deg(v) \leq 2s\omega$. Remove the rectangle corresponding to v from the representation and repeat the argument above to find a vertex that has degree at most $2s\omega(G)$ in the new representation. Continue until there are no vertices left. By definition, this shows that G is $2s\omega(G)$-degenerate, which completes the proof. ☐

Remark 2.3.19. *Given an s-sparse set of rectangles $\mathcal{R}$, then a good $2s\omega(G)+1$ colouring of the intersection graph G of $\mathcal{R}$ can be found in polynomial time. This is true as the argument to show that G is $(2s\omega(G)+1)$-colourable only uses that G is $2s\omega(G)$-degenerate. In fact, we have shown that given a rectangle intersection graph G, then any induced subgraph of G is $2s\omega(G)$-degenerate. Therefore, a $2s\omega(G)$-degenerate ordering of the vertices is computable in polynomial time as it suffices to search for any vertex v with $\deg(v) \leq 2s\omega(G)$ when looking for the next vertex in the ordering. Once we have this ordering, we can colour the vertices in reverse order using a greedy colouring, i.e., we can colour the vertices with the lowest available colour that has not been used on any of its neighbours so far. This can all be done in polynomial time.*

We now give a proof of a theorem by Asplund and Grünbaum [2], which states that $\chi(G) \leq 4(\omega(G))^2 - 3\omega(G)$ for a rectangle intersection graph G. Although the result is over 40 years old, the best improvement of this result is by Hendler [32], who has shown that $\chi(G) \leq 3(\omega(G))^2 - 2\omega(G) - 1$. The best known lower bound for general rectangle intersection graphs is $3\omega(G)$ (Kostochka [35]).

Theorem 2.3.20 (Asplund and Grünbaum [2])**.** *Let $G = (V, E)$ be a rectangle intersection graph, then we have $\chi(G) \leq 4(\omega(G))^2 - 3\omega(G)$.*

Proof. We partition the edge set of G into two sets, E_1 and E_2. Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$. We show the following:

1. $\chi(G_1) \leq \omega(G)$
2. $\chi(G_2) \leq 4\omega(G) - 3$

The partition is defined according to how the rectangles intersect. In general, there are four types of intersections that we distinguish between (see Figure 2.52). Namely, given an edge $e = (R, R')$:

- If one rectangle contains all 4 corners of the other it is a *containment* intersection.
- If one rectangle contains exactly 2 corners of the other it is an *overlap* intersection.
- If one rectangle contains exactly 1 corner of the other it is a *corner* intersection.
- If none of the corners of any of the rectangle are contained in the other, then the intersection is a *crossing* intersection.

Containment. Overlap. Corner. Crossing.

Figure 2.52.: Four different types of intersection.

Let $\mathbf{E_1} = \{e \in E : e \text{ is a crossing }\}$ and $\mathbf{E_2} = E \setminus E_1$.
Proof of (1). We show that G_1 is a comparability graph and therefore perfect (see Subsection 1.3.2). Given the representation of G_1 induced from the representation of G, we define the partial order $\leq_1$ on G_1 by

$$v_R \leq_1 v_{R'} \iff R \text{ is thinner in the horizontal direction than } R. \tag{2.4}$$

This is clearly a partial order and therefore G_1 is comparability graph and hence perfect. It follows that $\chi(G_1) = \omega(G_1) \leq \omega(G)$.
Proof of (2). To show that G_2 is $(4\omega(G) - 3)$-degenerate, we use an argument similar to the one in Lemma 2.3.18. We double count the number of ordered pairs (c, e) such that c is a corner of a rectangle in e that lies in the other rectangle in e. We assign to each rectangle the set of its 4 corners. The only changes to the proof of Lemma 2.3.18 are:

- Each corner can be involved in at most $\omega(G) - 1$ pairs. This is because the corner is also in the rectangle that it has been assigned to.

- Each edge e appears in at least 2 pairs because we don't have crossings in E_2.

Following the proof of Lemma 2.3.18 and taking these modifications into account, we obtain that $\chi(G_2) \leq 4\omega(G_2) - 3 \leq 4\omega(G) - 3$.
Let c_1 and c_2 be good colourings of G_1 and G_2 respectively. We construct a good colouring c of G by letting $c(v) = (c_1(v), c_2(v))$, i.e., each vertex receives one pair of colours. This is a good colouring of G because

$$e = (v, w) \in E_i \implies c_i(v) \neq c_i(w) \implies c(v) \neq c(w).$$

The number of colours c uses is the number of colours c_1 uses times the number of colours c_2 uses, which is at most $4(\omega(G))^2 - 3\omega(G)$. □

Note that rectangle representations that come from hook representations, do not have containment intersections. Regarding hook graphs as special rectangle intersection graphs we can adjust this proof to obtain a slightly better upper bound for hook graphs.

Proposition 2.3.21. *We have $\chi(G) \leq 3(\omega(G))^2 - 2\omega(G)$ for all hook graphs G.*

Proof. The proof is identical to the proof of Theorem 2.3.20. Here however, we don't need to assign all of the corners to the rectangles. This is because we don't have any intersection where the top-left hand corner of a rectangle is in the interior of another rectangle. Assigning the set of the other 3 corners to each rectangle and following the steps of the proof above delivers the upper bound of $3(\omega(G))^2 - 2\omega(G)$. □

After a more careful analysis of the structure of rectangle intersection graphs, the following result was proved by Chalersmook [10].

Theorem 2.3.22 (Chalermsook [10])**.** *$\chi(G) = O(\sigma(G)\omega(G)\log(\omega(G)))$, for all rectangle intersection graphs G, where $\sigma(G)$ is defined below.*

Definition 2.3.23. We define $\sigma(\mathcal{R})$ of a set $\mathcal{R}$ of rectangles in the plane as follows:

$$\sigma(\mathcal{R}) = \min\{d(\mathcal{R}), h(\mathcal{R})\} + 1,$$

where $d(\mathcal{R})$ and $h(\mathcal{R})$ are defined as follows:
Given a rectangle R, we define the *containment depth* of R, denoted $d(R)$, to be the number of rectangles $R' \in \mathcal{R}$ that contain R. We define $d(\mathcal{R})$ to be $\max\{d(R) : R \in \mathcal{R}\}$.
Let $h(\mathcal{R})$ be the size of a largest set of disjoint rectangles $\mathcal{I}$ satisfying:

- $\exists R \in \mathcal{R}$ such that $R' \subset R$ for all $R' \in \mathcal{I}$.

- There exists a horizontal line that intersects all rectangles in $R' \in \mathcal{I}$.

We define $\sigma(G)$ to be $\min\{\sigma(\mathcal{R}) : \mathcal{R}$ is a rectangle representation of $G\}$.

It is clear that all hook graphs G have $\sigma(G) = 1$ because a rectangle representation where the top-left corner of each rectangle lies on a unique point on the diagonal does not have containment intersections.

Corollary 2.3.24. $\chi(G) = O(\omega(G)\log(\omega(G)))$ *for all hook graphs G.*

In what follows, by a clique of rectangles, we mean a set of rectangles that pairwise intersect, and by an independent set of rectangles we mean a set of pairwise disjoint rectangles. We also say that a good colouring of rectangles in $\mathcal{R}$ is a colouring of the rectangles so that no two rectangles that intersect receive the same colour. We denote by $\omega(\mathcal{R})$, the size of the largest clique in $\mathcal{R}$. Before we sketch the proof of Theorem 2.3.22, we define three more parameters of a rectangle, which Chalermsook [10] used to help investigate the structure of rectangle representations.

Definition 2.3.25 ($\nu(R)$, η-coverage, τ-coverage)**.** Let $\mathcal{R}$ be a set of rectangles in the plane, and let rectangle $R \in \mathcal{R}$.

- Define $\mathcal{V}(R)$ to be the set of rectangles in $\mathcal{R} \setminus \{R\}$ that cross R and that are thinner than R in the horizontal direction. Let $\nu(R)$ be the size of the largest clique in $\mathcal{V}(R)$.
- Given $\eta \in \mathbb{N}$, an *η-coverage* of R is a clique $\mathcal{C}$ in $\mathcal{R} \setminus \{R\}$ such that we can partition $\mathcal{C}$ into two sets X_1 (resp. X_2), such that $|X_1| = |X_2| = \eta$ and all rectangles in X_1 and X_2 intersect the top (resp. bottom) boundary of R. We call X_1 and X_2 a *top* and a *bottom* coverage of R, respectively. $\eta(R)$ is the largest value of η for which we can find an η-coverage of R.
- Given $\tau \in \mathbb{N}$, a *τ-coverage* of R is defined to be a clique of size τ in $\mathcal{R} \setminus \{R\}$ whose rectangles contain both left corners of R or both right corners of R. We define $\tau(R)$ to be the largest value of τ for which there exists a τ-coverage of R.

Sketch of proof of Theorem 2.3.22. Let $\mathcal{R}$ be a rectangle representation of the graph G and let $\omega = \omega(\mathcal{R})$. We can assume that $h(\mathcal{R}) \le d(\mathcal{R})$. Otherwise we can decompose $\mathcal{R}$ into the sets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{d(\mathcal{R})}$, where $\mathcal{D}_i = \{R \in \mathcal{D} : d(R) = i\}$. In each set $\mathcal{D}_i$ we have no containment intersections and therefore we have $h(\mathcal{D}_i) = d(\mathcal{D}_i) = 0$ and $\sigma(\mathcal{D}_i) = 1$. Therefore the result for the case $h(\mathcal{R}) \le d(\mathcal{R})$ would imply the result for $h(\mathcal{R}) > d(\mathcal{R})$.
The algorithm to compute the colouring works using iterations. We let $\mathcal{R}_0 = \mathcal{R}$ and let β be a constant that is defined appropriately for the algorithm to work. Let $\omega = \omega(\mathcal{R})$ and $\sigma = \sigma(\mathcal{R})$. The input for each iteration is the set $\mathcal{R}_i$, together with the partition $\{S^i_j\}_{j=1}^{\beta 2^i}$ of $\mathcal{R}_i$. Where

$$S^i_j = \{R \in \mathcal{R}_i : (j-1)\frac{\omega}{\beta 2^i} \le \nu(R) < j\frac{\omega}{\beta 2^i}\}$$

with the additional property that $\omega(S^i_j) \le \frac{\omega}{2^i}$ for all j. The iteration returns a set $\mathcal{R}_{i+1} \subset \mathcal{R}_i$, together with a partition $\{S^{i+1}_j\}_{j=1}^{2^{i+1}}$ of $\mathcal{R}_{i+1}$ such that $\omega(S^{i+1}_j) \le \frac{\omega}{2^{i+1}}$. The rectangles that we remove are chosen so that we can colour them using $O(\omega\sigma)$ colours. This iteration is carried out k times, where $k = O(\log(\omega))$. We are left with the set $\mathcal{R}_k$, together with a partition of $\mathcal{R}_k$ into $O(\omega)$ sets S^k_j. By definition, $\omega(S^k_j) \le C$ for some constant C for all j. Applying Lemma 2.3.18 to each of these sets implies that we can find a $O(\omega)$-colouring of $\mathcal{R}_k$.

The iterations: For each rectangle $R \in S^i_j$, we begin the ith iteration by computing the size of the largest η-coverage and τ-coverage of R using rectangles in S^j_i. We denote these coverages by $\eta_i(R)$ and $\tau_i(R)$. We now show two important facts:

1. Given $m \in \mathbb{N}$ and a set $\mathcal{R}' \subset \mathcal{R}$. For $m \geq 2$, if all rectangles $R \in \mathcal{R}'$ satisfy $\eta(R) \leq m$, then we have $\omega(\mathcal{R}') \leq 3m$.

2. Given a rectangle R, and a clique $\mathcal{C}$ in $\mathcal{V}(R)$, then we have $\nu(R) \geq \min_{R' \in \mathcal{C}}\{\nu(R')\} + \lfloor \frac{|\mathcal{C}|}{2} \rfloor$.

The first property is easy to see: If we have that $\omega(\mathcal{R}') > 3m$, then consider a clique $\mathcal{C}$ of size $3m+1$. Remove the set $\mathcal{C}_{low}$ of $m+1$ rectangles from $\mathcal{C}$ whose top boundaries are the lowest. From the remaining rectangles $\mathcal{C} \setminus \mathcal{C}_{low}$, remove the set $\mathcal{C}_{high}$ of $m+1$ rectangles whose bottom boundaries are the highest. The set $\mathcal{C}_{high}$ (resp. $\mathcal{C}_{low}$) is a top (resp. bottom) coverage of any of the remaining rectangles (there is at least one rectangle remaining). Therefore, we have a rectangle R in $\mathcal{R}'$ that has $\eta(R) \geq m+1$.
The second property can be shown as follows: Consider a clique $\mathcal{C}$ in $\mathcal{V}(R)$. Then by there exists a point p that is contained in all rectangles in $\mathcal{C}$ by Remark 2.3.16. Remove the set $\mathcal{C}_{left}$ of $\lfloor \frac{|\mathcal{C}|}{2} \rfloor$ rectangles from $\mathcal{C}$ whose left boundary lie furthest to the left. Let $\mathcal{C}_{right}$ denote the set of remaining rectangles. Let R' be the rectangle in $\mathcal{C}_{right}$ whose right boundary is furthest to the left. Now consider a clique $\mathcal{C}'$ in $\mathcal{V}(R')$. As $R' \in \mathcal{V}(R)$, we have that $\mathcal{C}' \subset \mathcal{V}(R)$. Therefore, there must exist a point p' that is contained in all the rectangles in $\{R, R'\} \bigcup \mathcal{C}'$. If p lies to the left of p' then all the rectangles in $\mathcal{C}_{right}$ must contain p', in which case $\mathcal{C}_{right} \bigcup \mathcal{C}'$ is a clique in $\mathcal{V}(R)$. Similarly, if p' lies to the left of p, then $\mathcal{C}_{left} \bigcup \mathcal{C}' \bigcup \{R'\}$ is a clique in $\mathcal{V}(R)$. Therefore in either case, we have a clique in $\mathcal{V}(R)$ that contains at least $\min_{R' \in \mathcal{C}}\{\nu(R')\} + \lfloor \frac{|\mathcal{C}|}{2} \rfloor$ rectangles, which concludes the proof of the second property.
The ideas in the iterations are as follows:

- The maximum clique size in each set S_j^i is reduced by removing the set $\mathcal{F}_j^i$ of all rectangles in S_j^i that have large $\eta_i(R)$ (from Property 1).
- We partition $\mathcal{F}_j^i$ into 2 sets $\mathcal{A}_j^i$ and $\mathcal{B}_j^i$ that are 5-sparse and $O(\sigma)$-sparse, respectively. Therefore, as $\omega(S_j^i) \leq \frac{\omega}{2^i}$, we can apply Lemma 2.3.18 to find an $O(\frac{\sigma\omega}{2^i})$-colouring of $\mathcal{F}_j^i$. Using different sets of colours for each set $\mathcal{F}_j^i$ and summing over all j, we can colour the all the sets that have been removed using $O(\sigma\omega)$ colours.

The more delicate part is to choose sets $\mathcal{A}_j^i$ and $\mathcal{B}_j^i$ so that they are sparse. For all $R \in \mathcal{F}_j^i$, if we let the four corners of a rectangle R be members of the set of representative points of R, the only intersections that are unaccounted for are crossings
One idea is to add a point $p_R \in R$ that is contained in all the rectangles involved in a large η_i-coverage of R to the set of representative points of R. Consider another rectangle R' in $\mathcal{F}_j^i$ that crosses R and is wider than R. If p_R is in R' then the edge (R, R') is accounted for. If p_R is not in R', then suppose it lies below R' as in Figure 2.53. Any rectangle R^* in the

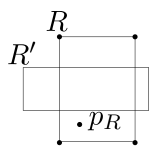


Figure 2.53.: An edge that is not accounted for by any of the corners or p_R.

large top coverage of R must satisfy at least one of the following statements:

- R^* intersects the two left corners of R'
- R^* intersects the two right corners of R'
- R^* crosses R'.

The set of all such rectangles R^* that cross R' forms a clique $\mathcal{C}_R^i$. A rectangle R^* in $\mathcal{C}_R^i$ is in S_j^i, which means that $\nu(R^*)$ is bounded below. Property 2 therefore implies that there cannot be too many rectangles in $\mathcal{C}_R^i$ as $\nu(R')$ is bounded above by definition of S_j^i. This is where the τ-coverage is used: If $\tau_i(R')$ is low, then $\mathcal{C}_R^i$ must contain many rectangles and therefore R must contain p_R and the edge (R, R') is accounted for. Chalermsook defines $\mathcal{A}_j^i$ to be the restriction of $\mathcal{F}_j^i$ to the set of rectangles R with sufficiently low value of $\tau_i(R)$ so that the point p_R is a representative point for any crossing intersection between R and a wider rectangle in $\mathcal{F}_j^i$. For all rectangle R in $\mathcal{A}_j^i$, the 4 corners of a rectangle R together with p_R are representation points of R in $\mathcal{A}_j^i$. Therefore $\mathcal{A}_j^i$ is 5-sparse. Note that the case where the point p_R lies above R' is argued similarly.

Sparseness of $\mathcal{B}_j^i$ The set $\mathcal{B}_j^i$ is just the set $\mathcal{F}_j^i \setminus \mathcal{A}_j^i$. When setting up $\mathcal{A}_j^i$ one can set the lower bound on $\eta_i(R)$ so that the upper bound on $\tau_i(R)$ for rectangles R in $\mathcal{A}_j^i$ is quite high. Therefore $\tau_i(R)$ for rectangles R in $\mathcal{B}_j^i$ is quite high. For a rectangle R in $\mathcal{B}_j^i$, we define $\mathcal{I}_R$ to be a maximal independent set in $\mathcal{V}(R) \bigcap \mathcal{B}_j^i$ that is minimal in the following sense: Given a rectangle R' in $\mathcal{V}(R) \bigcap \mathcal{B}_j^i$ that is not in $\mathcal{I}_R$, then there does not exist a rectangle $R_{\mathcal{I}} \in \mathcal{I}_R$ such that R' crosses $R_{\mathcal{I}}$, and R' is thinner than $R_{\mathcal{I}}$ in the horizontal direction. Define the set of representative points of a rectangle R in $\mathcal{B}_j^i$ to be the set of its corners, together with the points $\mathcal{P}_R$, where $\mathcal{P}_R$ is defined to be the set of intersection points between the boundaries of the rectangles in $\mathcal{I}_R$ and the boundary of R (see Figure 2.54). It is not difficult to see that

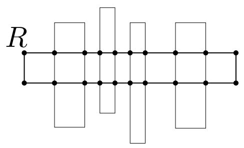


Figure 2.54.: The independent set $\mathcal{I}_R$ and the set of representative points of R.

the set $\mathcal{P}_R$ accounts for any edge (R, R'), where R' crosses R and R' is thinner than R in the horizontal direction. Therefore we have a valid set of representative points as the corners account for the rest of the crossings. Now if $|\mathcal{I}_R| = O(\sigma)$, we can apply Lemma 2.3.18 once again to show that $\mathcal{B}_j^i$ is $O(\sigma\omega)$-colourable and we have found the colouring. Chalermsook showed: If $h(\mathcal{R})$ is bounded, then an independent set $\mathcal{I}_R$ in $\mathcal{V}(R) \bigcap \mathcal{B}_j^i$ cannot contain too many rectangles.
He does this by showing that if $\mathcal{I}_R$ is too large, then there is a clique of size larger than $\frac{\omega}{2^i}$ in S_j^i, which is a contradiction of all the sets being in S_j^i. This clique consists of some rectangles in the τ-coverages of $R' \in \mathcal{I}_R$. To find this clique he argues as follows: Given $R' \in \mathcal{I}_R$, the value $\tau_i(R')$ is large enough so that one can use Property 2, together with the definition of $\{S_j^i\}_{j=1}^{\beta 2^i}$, to find many rectangles R^* in a τ-coverage of R' that intersects the two left corners of R or the two right corners of R. The bound on $h(\mathcal{R})$ implies that such a rectangle R^* cannot be in the τ-coverages of many different rectangles in $\mathcal{I}$. If the independent set is too

large then we obtain a clique that is larger than the upper bound on the maximum clique in each of the sets S^i_j. One concludes that the size of $\mathcal{I}_R$ is bounded and that $\mathcal{B}^i_j$ is $O(\sigma)$-sparse and hence $O(\sigma\omega)$-colourable. The output is the set $\mathcal{R}_{i+1}$ of uncoloured rectangles remaining, together with the partition $\{S^i_j\}_{j=1}^{\beta 2^{i+1}}$ of $\mathcal{R}_{i+1}$, where

$$S^i_j = \{R \in \mathcal{R}_i : (j-1)\frac{\omega}{\beta 2^{i+1}} \leq \nu(R) < j\frac{\omega}{\beta 2^{i+1}}\}.$$

This concludes the description of the iterations and the sketch of the proof that $\chi(\mathcal{R}) = O(\sigma(G)\omega(G)\log(\omega(G)))$ for all rectangle intersection graphs G. □

Remark 2.3.26. *One can find a $O(\sigma(G)\omega(G)\log(\omega(G)))$ colouring in polynomial time given the rectangle representation of the graph G.*

Proof. First note that the number of maximal cliques in a rectangle intersection graph is quadratic in the number of vertices and these can be computed in polynomial time (see Asano and Imai [3]). It follows that in the *ith* iteration, computing the values of $\eta_i(R)$ and $\tau_i(R)$ can be done in polynomial time. It also follows that the value of $\nu(R)$ at the beginning of the algorithm can be calculated in polynomial time. Therefore partitioning the vertices into the sets $\{S^i_j\}_{j=1}^{\beta 2^{i+1}}$ and computing the sets $\mathcal{A}^i_j$ and $\mathcal{B}^i_j$ can all be done in polynomial time. Computing the good colourings of $\mathcal{A}^i_j$ and $\mathcal{B}^i_j$ can be done in $O(n)$ time, because of Lemma 2.3.18. □

It is widely believed that a linear upper bound exists for the chromatic number of rectangle intersection graphs; however, even in Chalermsook's result, $\tau(G)$ can be linear in terms of $\omega(G)$. Therefore no improvement on the quadratic upper bound can be deduced from this bound for rectangle intersection graphs in general. Although we have an extra restriction, we have not been able to improve this bound for hook graphs in general. We have made an improvement in the case of triangle-free hook graphs, which we will explain later. In what follows, we denote the cycle of length n by C_n and we show that $\overline{C_n}$ is not a hook graph for $n \geq 7$. The motivation behind this result is the Strong Perfect Graph Theorem (see Subsection 1.3.2).

Proposition 2.3.27. *$\overline{C_n}$ is not a hook graph for $n \geq 7$.*

In the proof the reader should note that although we show that $\overline{C_n}$ is not a hook graph, we use C_n in the argumentation.

Proof. Let $n \geq 7$ and assume that $\overline{C_n}$ is a hook graph. Let v_1 be the minimum vertex in a hook ordering of $\overline{C_n}$ and let v_m be its highest neighbour in the hook ordering. Because there are at least 7 vertices in C_n there exists a path v_m, v_i, v_j, v_k of length 3 in C_n starting at v_m that does not contain v_1 (see Figure 2.55).

Because of the cross completion property on the vertices v_1, v_i, v_j, v_m, we must have that $v_i <_h v_j$, and hence the ordering of these four points must be as shown in Figure 2.56. When trying put v_k into the ordering, we must violate the cross completion property:

- If $v_1 <_h v_k <_h v_j$, we have a contradiction of the cross completion property as (v_1, v_j) and (v_k, v_m) are both edges of $\overline{C_n}$, but (v_j, v_k) is not.

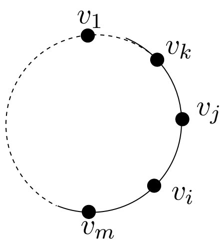


Figure 2.55.: Path from v_m without v_1.

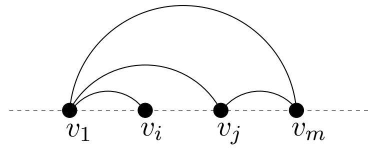


Figure 2.56.: The order of v_1, v_i, v_j, v_m.

- If $v_j <_h v_k$, we have a contradiction of the cross completion property as (v_i, v_k) and (v_1, v_j) are both edges of $\overline{C_n}$, but (v_j, v_k) is not.

This shows that it is not possible to find an ordering of the vertices of $\overline{C_n}$ that satisfies the cross completion property, hence $\overline{C_n}$ is not a hook graph. □

For $n \leq 6$, $\overline{C_n}$ is a hook graph. Figure 2.57 shows a valid hook ordering of $\overline{C_6}$.

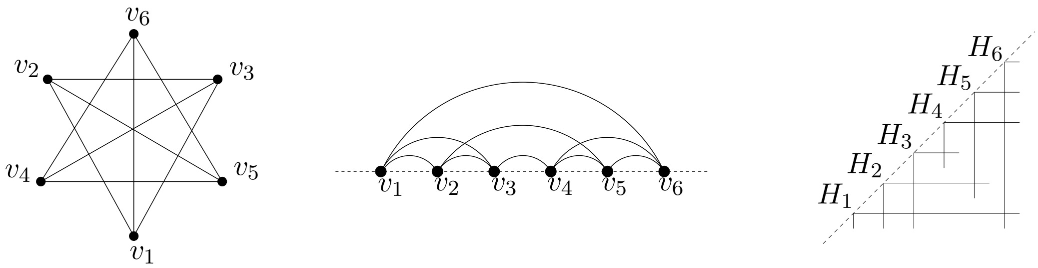


Figure 2.57.: A hook ordering of $\overline{C_6}$, its ordering on the real line, and its hook representation.

We have shown that the only odd antihole (see Theorem 1.3.17) that a hook graph can contain is $\overline{C_5}$, which is isomorphic to C_5. We now describe all possible different hook orderings of cycles, and therefore odd holes.

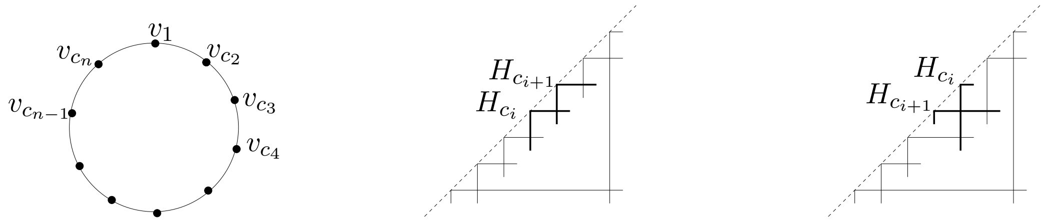
Theorem 2.3.28. *Given a representation of a cycle C_n, consider the hook ordering as a permutation $\pi_h = (v_1 v_2 \ldots v_n)$. Let $\pi_c = (v_1 v_{c_2} \ldots v_{c_n})$ be the permutation of the vertices that we read when going around the cycle C_n in the direction where v_{c_2} is the lowest neighbour of v_1 in the hook ordering (see the left of Figure 2.58). Then the following statements hold:*

1. *$v_{c_n} = v_n$, and*
2. *π_h can be obtained from π_c by a sequence of transpositions of neighbouring pairs of vertices $(v_{c_i}, v_{c_{i+1}})$ with $2 \leq i \leq n-2$ such that no vertex is in more than one transposition.*

Proof of 1. This is immediate from the observation that given $i \leq n-2$, then

$$\text{if } v_1 <_h v_{c_i} <_h v_{c_n} \text{ then } v_1 <_h v_{c_{i+1}} <_h v_{c_n}.$$

If this were not true, then $(v_1, v_{c_i}, v_{c_n}, v_{c_{i+1}})$ would violate the cross completion property. Therefore, as $v_1 <_h v_{c_2} <_h v_{c_n}$, we can apply the observation above and conclude that the



The permutation π_c. When $\pi_h = \pi_c$. $(v_{c_i}, v_{c_{i+1}})$ transposed.

Figure 2.58.: π_c, its representation, and a transposition.

first statement holds.
Proof of 2. We first prove the following:

$$v_{c_{i-1}} <_h v_{c_{i+1}} \text{ for } 2 \leq i \leq n-1. \tag{2.5}$$

Suppose not, then the path $P = v_1, v_{c_2}, \ldots, v_{c_{i-1}}$ and the path $P' = v_{c_{i+1}}, v_{c_{i+2}}, \ldots, v_n$ alternate (see Lemma 2.2.8), we get that an edge of P must intersect an edge of P' in the hook ordering on a line. The only edge between P and P' is (v_1, v_n), hence P' must intersect this edge to avoid violating the cross completion property. However, v_1 and v_n are the largest and smallest vertices in the hook ordering. Therefore it is not possible for P' to cross (v_1, v_n) and Equation 2.5 must hold.
We now show that Equation 2.5 implies that we can obtain π_h from π_c using a sequence of transpositions as stated above. Consider the position x of v_{c_i} in π_h for $2 \leq i \leq n-1$. Then $|i - x| \leq 1$, else we would have a contradiction of Equation 2.5:
In the case where $x = i + 1$, then the vertex in position i in π_h must be $v_{c_{i+1}}$: Indeed, there must be a vertex v_{c_k} with $k > i$ whose position y in π_h satisfies $y \leq i$. By Equation 2.5, the only vertex that can be below v_{c_i} with respect to $<_h$ is $v_{c_{i+1}}$. If $y \leq i - 1$ then we must find a vertex v_{c_l} with $l \leq i - 1$ which is larger than y in the hook ordering, which contradicts Equation 2.5. This argument is identical if $x = i - 1$
We can go through all the vertices v_{c_i} in π_c in increasing order and transpose them when necessary to their final position x in π_h. Whilst doing this, we maintain the invariant that a vertex has been seen at most once, and any vertex that has been seen is in its final position. Let v_{c_i} be a vertex that has not yet been seen. If $i = x$ then there is nothing to do. Otherwise, if $x = i + 1$ then the position of vertex $v_{c_{i+1}}$ must be i and hence $v_{c_{i+1}}$ can't have been seen yet. In this case, we can transpose both vertices to obtain their final position. We then label v_{c_i} and $v_{c_{i+1}}$ as seen. We can continue this process until all the vertices are in their final positions. By the invariant maintained, we have found a sequence of transpositions with the properties wanted. □

Remark 2.3.29. *Given a cycle C_n the following orderings of the vertices are all the valid hook orderings of C_n. Let v_{c_1} be a vertex in C_n and let $\pi_c = (v_{c_1} v_{c_2} \ldots v_{c_n})$ be the permutation of the vertices that we encounter when going around the cycle. Let π be a permutation $(v_{c_1} v_2 \ldots v_{n-1} v_{c_n})$ that can be obtained via a sequence of transpositions as in 2 of Proposition 2.3.28, then π is a valid hook ordering.*

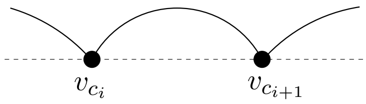


Figure 2.59.: Locally before a transposition.

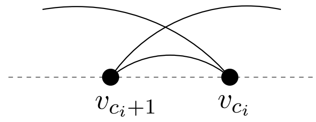


Figure 2.60.: Locally after a transposition.

Proof. Figure 2.58 shows that π_c is a valid hook ordering. We perform the transpositions one by one and show that the permutation is a valid hook ordering. Consider the hook ordering on a line, then before we transpose $(v_{c_i}, v_{c_{i+1}})$ the picture looks locally like Figure 2.59 as v_{c_i} and $v_{c_{i+1}}$ have not been involved in a transposition. Figure 2.60 shows how the ordering would be locally after transposing them. This is still a valid hook ordering because the only new intersection between edges does not violate the cross completion property as $(v_{c_i}, c_{c_{i+1}}) \in E$ and therefore we still have a valid hook ordering. $\square$

Now we discuss the case $\omega(G) = 3$. Asplund and Grünbaum [2] showed that the chromatic number of triangle-free rectangle intersection graphs is at most 6. They also gave an example that achieves this upper bound. We use the structure of the representations of cycles to show that triangle free hook graphs can be coloured with at most 4 colours.

Theorem 2.3.30 (Triangle-free hook graphs)**.** *If G triangle-free hook graph, then $\chi(G) \leq 4$.*

Proof. We will construct a partition of the vertex set V into 2 sets, V_1 and V_2, such that $G|_{V_1}$ and $G|_{V_2}$ are both trees. Then we can find good 2-colourings c_1 and c_2 of $G|_{V_1}$ and $G|_{V_2}$ respectively, since trees are bipartite. Let the set of colours that c_1 uses be disjoint to the set that c_2 uses. The following colouring is a good colouring of G with 4 colours.

$$c(v) = \begin{cases} c_1(v) & \text{if } v \in V_1. \\ c_2(v) & \text{if } v \in V_2. \end{cases}$$

Before we can state our partition, we need to do a little work.

Definition 2.3.31. Given a hook representation of a graph G, let $e = (u, w)$ be an edge, and v a vertex such that $u <_h v <_h w$. We say that v *leaves e to the south* if (u, v) is an edge (see Figure 2.61). Similarly, we say that v *leaves e to the east* if (v, w) is an edge (see Figure 2.62).

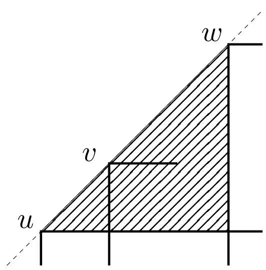


Figure 2.61.: v leaving (u, w) to the south.

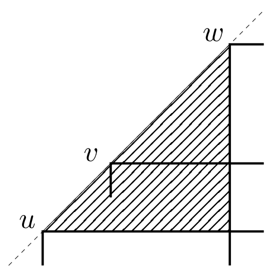


Figure 2.62.: v leaving (u, w) to the east.

Note that a vertex cannot leave an edge e both to the south and to the east in a triangle-free hook graph. In fact, a stronger statement also holds:

Lemma 2.3.32. *Given a triangle-free hook representation, if a vertex v leaves an edge e to the east, it cannot leave another edge e' to the south.*

Proof. Let $e = (u, w)$ and $e' = (u', w')$ such that v leaves e to the east and e' to the south. We first note that $u \neq u'$, and $w \neq w'$, otherwise (u, v, w) or (u', v, w') form triangles.

1. We have that $u <_h u'$, otherwise we have $u' <_h u <_h v <_h w$, which implies that (u, v) (cross completion property), and thus (u, v, w) forms a triangle.
2. Analogously, we have that $w <_h w'$, otherwise we have $u' <_h v <_h w' <_h w$ implies that (u', v') is an edge (cross completion property), and thus (u', v, w') forms a triangle.

Items 1 and 2 imply that $u <_h u' <_h w <_h w'$. However, the cross completion property now implies that (u', w) is an edge, and therefore (u', v, w) forms a triangle. This is a contradiction to the graph being triangle-free. We have shown that if a vertex leaves an edge to the east, then it does not leave an edge to the south. □

Now we are ready to define the sets V_1 and V_2. Given a triangle-free hook representation, let v be in V_1 if there exists an edge e such that v leaves e to the south. Let $V_2 = V \setminus V_1$. We claim that $G|_{V_1}$ and $G|_{V_2}$ are trees. Consider the hook representation of a cycle C in G. Let v_s and v_l be the smallest and the largest vertices in the hook ordering of C, respectively, then by Proposition 2.3.28, we have $(v_s, v_l) \in E$. Therefore by definition, the other neighbour $v_{s'}$ of v_s in the cycle leaves (v_s, v_l) to the south. In addition, the other neighbour $v_{l'}$ of v_l must leave (v_s, v_l) to the east. Lemma 2.3.32 implies that $v_{l'}$ cannot leave any edge to the south. Therefore $v_{l'} \in V_2$ and $v_{s'} \in V_1$ by definition. Therefore, none of the cycles in G can be completely contained in one of the vertex sets V_1 or V_2 and we are done. □

The question remains open whether all triangle-free hook graphs are 3-colourable or not. The odd cycle is a triangle-free hook graph, which requires 3 different colours. Theorem 2.3.30 shows a difference between triangle-free rectangle intersection graphs and triangle-free hook graphs. We hope that the techniques used in the proof can be extended to graphs whose clique number is larger than 2, but we are yet to succeed in generalising the method.

2.3.4. Approximating the Clique Covering Number

Given a graph G, a *clique covering* is a partition of the vertex set into subsets, such that G restricted to each subset is a clique. The *clique covering number* $\gamma(G)$ is the size of a smallest clique covering, i.e., the minimum number of subsets in the partition. Equivalently, it is the colouring number of the complement of G ($\chi(\overline{G})$). For segment graphs, we already mentioned in Subsection 2.1.1 that computing $\gamma(G)$ is NP-hard as a consequence of planar graphs being a subclass of segment graphs (Chalopin and Gonçalves [13]). For rectangle intersection graphs, Asano and Imai [3] have shown that it is NP-hard to compute $\gamma(G)$. However, for Interval graphs (see Algorithm 6), 2DORGS, and outerplanar graphs, computing $\gamma(G)$ can be done in polynomial time. For hook graphs, the complexity of calculating $\gamma(G)$ remains unknown. In this subsection we give an algorithm that computes a 2-approximation of the clique covering number in $O(n^2)$ time. In doing this, we show that the independence number $\alpha(G)$ is a 2-approximation of $\gamma(G)$ for hook graphs. For general rectangle intersection graphs, the problem of trying to bound $\gamma(G)$ by a linear function of $\alpha(G)$ dates back to 1965, where Wegner [60] asked whether one could bound $\gamma(G)$ by $2\alpha(G) - 1$. Since then there has been much interest in trying to find a linear upper bound. The best upper bound that is known is by Károlyi [33], who shows that $\gamma(G) \leq \alpha(G)\log(\gamma(G)) + 2$. Fon-Der-Flaass and Kostochka [24], show that there exists a rectangle representation of a graph G, where $\gamma(G) \geq \frac{5}{3}\alpha(G)$. For hook graphs, we can apply a result by Chepoi1 and Felsner [15] to show that $\gamma(G) \leq 8\alpha(G)$ for all hook graphs G: Chepoi1 and Felsner [15] show that $\gamma(G) \leq 8\alpha(G)$ for a rectangle intersection graph G with the following property: There exists a rectangle representation of G that is pierced by an axis-monotone curve C, i.e., all rectangles have their top left corner above C and their bottom right corner below C. Where, an *axis-monotone curve* is an unbounded Jordan curve C, such that the intersection of C with any horizontal of vertical line is a segment (this includes points). The diagonal in a hook representation is an example of an axis-monotone curve. Therefore, given a rectangle representation that comes from a hook representation, we can translate the diagonal to the right by a very small amount to obtain a rectangle representation that is pierced by an axis monotone curve. Giving us $\gamma(G) \leq 8\alpha(G)$. The ideas used to obtain the 2-approximation in this subsection are similar to the ideas used by Chepoi1 and Felsner [15]. Before we state the algorithm that gives the 2-approximation for hook graphs, we introduce the notion of a hitting set, which is closely related to a clique covering of $\mathcal{R}$.

Definition 2.3.33. Given a set of rectangles $\mathcal{R}$, we define a *hitting set* of $\mathcal{R}$ to be a set P of points in the plane such that every rectangle $R \in \mathcal{R}$ contains at least one point p in P.

Now as the set of rectangles that intersect a single point must be a clique, we can obtain a clique covering of size k from a hitting set of $\mathcal{R}$ of size k. From a consequence of Helly's Theorem (See Remark 2.3.16) we can also obtain a hitting set of size k given a clique covering of size k. Therefore, the clique covering number is equal to the size of the smallest hitting set of a rectangle intersection representation of G.

Theorem 2.3.34. *Given a hook ordering of a hook graph:*

1. *There is a clique covering of size at most* $2\alpha(G)$.
2. *We can calculate such a clique covering in* $O(n^2)$ *time.*

This theorem implies that we have a quadratic time algorithm to compute a 2-approximation of the clique covering number $\gamma(G)$ (and the independence number $\alpha(G)$) as
$\alpha(G) \leq \gamma(G) \leq |OUTPUT| \leq 2\alpha(G) \leq 2\gamma(G)$.

Proof of (1). We define the *greedy independent set* $\mathcal{I}$ with respect to a hook ordering to be the set that is constructed as follows: Let $\mathcal{I}$ start as the empty set, then go through the vertices in increasing order with respect to $<_h$. When we reach vertex v, we add it to $\mathcal{I}$ if and only if it is not adjacent to any vertices that are already in $\mathcal{I}$. Figure 2.63 shows the hooks in a greedy independent set, whose vertices are marked in black. Given a greedy independent set $\mathcal{I} = \{v_{\beta_1}, v_{\beta_2}, \ldots, v_{\beta_n}\}$, we define $\mathcal{P}$ to be the set of points $\{P_i : 1 \leq i \leq n-1\}$, where $P_i = (\beta_{i+1} + \epsilon, \beta_i - \epsilon)$, for $0 < \epsilon < \frac{1}{2}$ (see Figure 2.64). Take the rectangle representation of

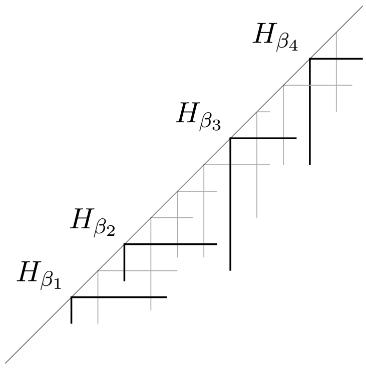


Figure 2.63.: The greedy independent set.

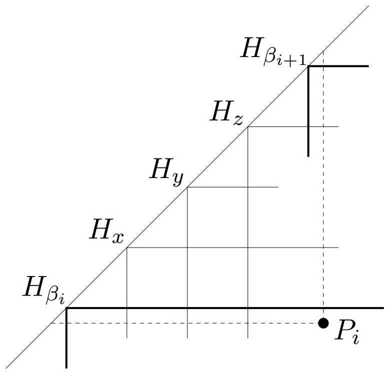


Figure 2.64.: A closer look between two hooks in $\mathcal{I}$.

G that corresponds to the canonical hook representation. We define $\mathcal{A}$ to be the set of all vertices whose rectangles contain at least one point in $\mathcal{P}$. Define $\mathcal{B}$ to be $V \backslash \mathcal{A}$. In Figure 2.64, the vertices that correspond to the hooks H_{β_i}, H_x, and H_z are in $\mathcal{A}$ and the other vertices are in $\mathcal{B}$. By definition, the set $\mathcal{P}$ is a hitting set of $\mathcal{A}$. In addition, $|\mathcal{P}| = |\mathcal{I}| - 1 < |\alpha(G)|$. Therefore, we have that $\gamma(G|_{\mathcal{A}}) < \alpha(G)$. We now need to bound $\gamma(G|_{\mathcal{B}})$, which can be done using the following remark.

Remark 2.3.35. *The hook graph restricted to the set $\mathcal{B}$ is an interval graph.*

Proof (of Remark 2.3.35). For all $H_k \in \mathcal{B}$, there exists an vertex v_{β_j} in the greedy independent set such that $\beta_j \leq k < \beta_{j+1}$. With β_j defined like this, we have the following:

1. H_k intersects the horizontal line $y = \beta_j$:
   If $v_k = v_{\beta_j}$, the statement is obvious. If $v_k \neq v_{\beta_j}$ then, by definition of $\mathcal{I}$, there exists a vertex $v_l \in \mathcal{I}$ with $l \leq \beta_j$ such that $(v_k, v_l) \in E$.

2. H_k does not cross the vertical line $x = \beta_{j+1}$:
   Indeed, as H_k must cross the line $y = \beta_j$ by the first property, if it also crosses the vertical line $x = \beta_{j+1}$ then its corresponding rectangle would contain the point P_i, which would imply H_k is in $\mathcal{A}$. Note that we use the fact that we have a canonical representation, i.e., if two hooks intersect, then they intersect in a crossing that is defined so that the rectangles would contain P_i.

From the first property we get that the hooks H_k in $\mathcal{B}$ with $\beta_j \leq k < \beta_{j+1}$ form an interval graph G_j. The second property implies that all these interval graphs are disjoint, i.e., there are no edges between G_j and G'_j for $j \neq j'$. As $\mathcal{B} = \bigcup_j G_j$, and the disjoint union of interval graphs are interval graphs, we have that $G|_{\mathcal{B}}$ is an interval graph. $\square$

We know that interval graphs are perfect, and therefore $\gamma(G|_{\mathcal{B}}) = \alpha(G|_{\mathcal{B}}) \leq \alpha(G)$. An optimal clique covering of $\mathcal{B}$ together with the clique covering of $\mathcal{A}$ is a clique covering $\mathcal{C}$ of G that satisfies $|\mathcal{C}| \leq 2\alpha(G)$ $\square$

Proof of (2). Here is an outline of a quadratic time algorithm that computes the clique covering described above.

Algorithm 5 CliqueCoveringApproximation

Input: A hook graph $G = (V, E)$ together with a hook ordering of the vertices $v_1, v_2, .., v_n$
Output: clique covering of size at most $2\alpha(G)$

1: Compute the canonical indexed interval representation $I_1, I_2, \ldots, I_n$ of this hook ordering

2: Compute the greedy independent set $I_{\beta_1}, I_{\beta_2}, \ldots, I_{\beta_n}$ and the points P_i
3: Compute the sets $\mathcal{A}$ and $\mathcal{B}$
4: $\mathcal{C}_{\mathcal{A}} \leftarrow$ clique covering of $\mathcal{A}$
5: Compute an interval representation of $\mathcal{B}$
6: $\mathcal{C}_{\mathcal{B}} \leftarrow$ clique covering of $\mathcal{B}$
7: **return** $\mathcal{C}_{\mathcal{A}} \bigcup \mathcal{C}_{\mathcal{B}}$

Line 1: We start by finding the canonical indexed interval representation of G and its corresponding rectangle representation, which can be done in quadratic time. This operation is not necessary, but it clarifies the explanation.
Line 2: We then compute $\mathcal{I}$ and $\mathcal{P}$ simultaneously in quadratic time. More precisely, when we process a vertex, we just check its adjacencies with the current members of $\mathcal{I}$ in linear time. If it is added to $\mathcal{I}$ we add the appropriate point to $\mathcal{P}$, which can be computed in constant time.
Line 3: For each i, we check for each vertex v_k with $\beta_i \leq k < \beta_{i+1}$ whether its rectangle R_k contains any of the points P_i and P_{i-1} in $\mathcal{P}$. Note that it is enough to check this because R_k cannot contain any other point in $\mathcal{P}$. (β_{i+1}, β_i) is contained in R_k if and only if the indexed interval I_k contains both β_{i+1} and β_i. Because we have already computed the indexed intervals, this can be checked in constant time.
Therefore this line can be carried out in linear time.
Line 4: This is immediate and can be stored whilst calculating $\mathcal{A}$.
Line 5: From the proof of 1, the graph $G|_{\mathcal{B}}$ is an interval graph. From the hook representation, we can project the horizontal parts of each hook to obtain the interval representation. Similarly to finding and storing the upper-neighbourhoods of a vertex in HookWeightedMaxClique, we find the interval representation and store it as an ordered list of endpoints.
Line 6: Once we have the interval representation, we can use the algorithm stated below to give us an optimal clique covering of $\mathcal{B}$ in linear time. The last line is computed in constant time. This concludes the proof that CliqueCoveringApproximation is an $O(n^2)$ time algorithm. □

Remark 2.3.36. *Given an interval representation of a graph, we can compute a clique covering in linear time given the list of endpoints in increasing order as an input.*

Proof. The algorithm below computes the clique covering in linear time.

Basically the algorithm goes through the endpoints in increasing order, finding the elements of the clique one after another. After initialising all the variables, it adds vertices to the clique C_1 (Line 8) until it meets a right endpoint. When it meets a right endpoint it checks if the vertex is in C_1, in which case the clique C_1 is stored and we proceed to in the same way with the next clique, which is initialised in Lines 10 and 11. The set A is not

Algorithm 6 IntCliqueCovering

Input: List of endpoints of the intervals in a representation of graph G in increasing order $x_1, x_2 \ldots x_{2n}$

Output: A minimum clique covering of G

1: $i \leftarrow 1$
2: $C_1 \leftarrow \emptyset$
3: $X \leftarrow \emptyset$
4: $A \leftarrow \emptyset$
5: **for** $j = 1, 2, \ldots 2n$ **do**
6:     $v \leftarrow$ vertex whose interval has endpoint x_j
7:     **if** x_j is a left endpoint of v **then**
8:         $C_i \leftarrow C_i \bigcup \{v\}$
9:     **if** x_j is a right endpoint and v is in C_i **then**
10:         $i \leftarrow i + 1$
11:         $C_i \leftarrow \emptyset$
12:         $A \leftarrow A \bigcup \{v\}$
13: **return** $\{C_1, C_2, \ldots, C_{i-1}\}$

necessary in the algorithm, but we included it just to help argue that the clique covering is optimal. By construction, A is an independent set (Line 12) that has the same cardinality as the clique covering $C_1, C_2, \ldots, C_{i-1}$ that we obtain. As the largest independent set cannot be larger than the smallest clique covering, we must have an optimal clique covering. This is a linear time algorithm as all the operations in the FOR LOOP can be carried out in constant time. Note that Line 9 can be carried out in constant time as we can we can keep track of which clique a vertex is in at the time we meet it. As this is repeated $2n$ times, the entire algorithm takes $O(n)$ time. □

3. A Problem Related to Assigning Frequencies

In this chapter we focus on a colouring problem that is related to conflict-free colourings of point sets in the plane. Conflict-free colourings were first introduced by Even et al. [21], who were motivated by a frequency assignment problem in wireless networks. In this chapter, we investigate the chromatic number of hypergraphs whose vertices are a finite set of points in $\mathbb{R}^2$ and whose edges are induced by axis-aligned rectangles that all intersect a horizontal line. We also study the question in the restricted setting, where edges are induced by axis-aligned rectangles that contain at least k points. More specifically, we prove that some bounds given by Keszegh [34] on the number of colours required for a good colouring of any point set are tight. This is achieved by constructing point sets, whose range spaces require a sufficient number of colours for a good colouring. Regarding the problem that we study, only one case remains unresolved. Namely, finding the least upper bound on the chromatic number of such hypergraphs where each edge contains at least 6 vertices.

In Section 3.1 we begin with a motivation of the problem. We then introduce the main definitions and explain the important relation between the conflict-free colouring number and the chromatic number mentioned above. We conclude the section with an overview of related results. In Section 3.2 we present our results about hypergraphs induced by axis-aligned rectangles that are all pierced by a horizontal line. We also include the proofs by Keszegh [34] that give upper bounds on the chromatic number of such hypergraphs.

3.1. Definitions and Motivation

In this section we first introduce some important notions. We then give an overview of some related problems in the literature (see Subsection 3.1.3).

3.1.1. A Frequency Assignment Problem

First we discuss the frequency assignment problem in wireless networks, which motivated Even et al. [21] to introduce conflict-free colourings. We are given a finite number of fixed base stations in the plane, each of which is assigned a (not necessarily unique) frequency over which it communicates. We assume that if a base station is within range of a client, then they can communicate. The mobile phone of a client continuously scans for frequencies of base stations that lie within its range. For the signal to be clear, there must be a base station within range that communicates over a frequency that is different to all the other frequencies of the base stations that are also within range (because of interference). Assume that the base stations are already situated so that there is always a base station within range of a client. We would like to assign frequencies to the stations so that the client has a clear signal everywhere. A simple solution to this problem is to assign different frequencies to all the base stations. However, we would like to use as few frequencies as possible to maximize the differences between the frequencies used and to minimize costs. To translate this to a graph theoretical problem, we define the following hypergraph.

Definition 3.1.1. Let P be a finite point set in the plane and let $\mathcal{F}$ be a set of regions in the plane. We define the following hypergraph:

$$G(P,\mathcal{F}) = (V,\mathcal{E}), \text{ where } V = P \text{ and } \mathcal{E} = \{f \cap P : \ f \in \mathcal{F}, \ |P \cap f| \geq 2\}.$$

We say that a region $R \in \mathcal{F}$ *induces* an edge $e \in \mathcal{E}$ if $e \in R \cap P$.

In the literature $G(P,\mathcal{F})$ is sometimes called a *range space* and denoted $(P,\mathcal{F})$. To model the frequency assignment problem, we let P be the positions of the stations in the plane. We let $\mathcal{F}$ be the set of all regions R_x, where R_x is the range of a client when positioned at point x in the plane. In this case, a good frequency assignment is equivalent to a conflict-free colouring of the vertices of $G(P,\mathcal{F})$, where a conflict-free colouring is defined as follows.

Definition 3.1.2. Given a hypergraph $G = (V,\mathcal{E})$, a *conflict-free colouring* of G is a colouring $c : V \to [n]$, such that for all edges $e \in \mathcal{E}$, there exists a vertex $v \in e$ for which $c(v) \neq c(w)$ for all $w \in e \setminus \{v\}$. The *conflict-free colouring number* of G, denoted by $cf(G)$, is defined to be the minimum number of colours needed in a conflict-free colouring of G.

The frequency assignment problem corresponds to a conflict-free colouring of $G(P,\mathcal{F})$. The minimum number of frequencies needed is the conflict-free colouring number of $G(P,\mathcal{F})$. The frequency assignment problem can also be seen in the dual setting, that is, by considering the station's ranges: For a point p, define $r(p)$ to be the set of stations whose range contains p. The frequency assignment problem corresponds to assigning a frequencies to stations, so that for all $p \in \mathbb{R}^2$ there exists a station s in $r(p)$ who has been assigned a different frequency to all the frequencies of other stations in $r(p)$. This can also be modeled by finding a conflict-free colouring of a hypergraph G whose vertices are the stations and whose edge set is the set $\{r(p) : p \text{ is a station}\}$. This problem has also been the focus of much research; however, in this thesis, we will not deal with the dual setting further. We refer the interested reader to a survey by Smorodinsky [57] for further reading about the dual problem.

3.1.2. Conflict-Free Colourings

Given a family $\mathcal{F}$ of regions in the plane, Even et al. investigated the least upper bound on the conflict-free colouring number of $G(P,\mathcal{F})$ for any point set P of size n.

Definition 3.1.3. Given $n \in \mathbb{N}$ and a fixed set of regions $\mathcal{F}$, we define $cf(n,\mathcal{F})$ to be the minimum number of colours c so that we can find a conflict-free colouring of any point set of size n with c colours.

Remark 3.1.4.

$$cf(n,\mathcal{F}) = \max\{cf(G(P,\mathcal{F})) : \ P \subset \mathbb{R}^2, \ |P| = n\}.$$

In this chapter, we will only consider the conflict-free colouring number of graphs induced by points and families in the plane as described above. Unless stated otherwise, all points and regions that we mention are in the plane.

Note that the set $\mathcal{F}$ often has infinitely many elements, but the graph $G(P, \mathcal{F})$ is always finite. This is because many regions in $\mathcal{F}$ can induce the same edge. In fact, in this thesis we will only consider infinite families of regions. Three families that have been the subject of much study are given below.

- For real numbers $a, b, c, d \in \mathbb{R}$, an *axis-aligned rectangle* is a set of the form $\{(x, y) \in \mathbb{R}^2 : a < x < b, c < y < d\}$. We let $\mathcal{R}$ denote the set of all axis-aligned rectangles.
- For real numbers $a, b, c \in \mathbb{R}$, a *disc* is a set of the form $\{(x, y) \in \mathbb{R}^2 : (x - a)^2 + (y - b)^2 < c\}$. We let $\mathcal{D}$ denote the set of all discs.
- For real numbers $a, b, c \in \mathbb{R}$, a *bottomless rectangle* is a set of the form $\{(x, y) \in \mathbb{R}^2 : a < x < b, y < c\}$. We let $\mathcal{B}$ denote the set of all bottomless rectangles.

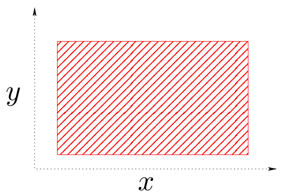


Figure 3.1.: An axis-aligned rectangle.

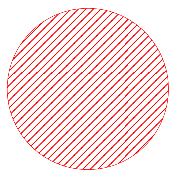


Figure 3.2.: A disc.

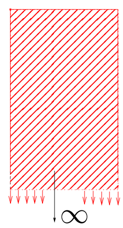


Figure 3.3.: A bottomless rectangle.

Given the point set P shown in Figure 3.4 the three bottomless rectangles in the figure imply that $\{x, y\}$, $\{x, z\}$ and $\{x, y, z\}$ are all edges of $G(P, \mathcal{B})$. Here, $\{x, z\}$ is not an edge of $G(P, \mathcal{B})$ as every bottomless rectangle containing x and z must contain y.

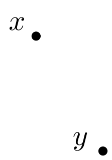


Figure 3.4.: Point set P and some bottomless rectangles.

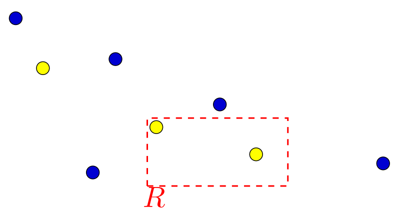


Figure 3.5.: A bad colouring of $G(P, \mathcal{R})$.

Figure 3.5 shows a 2-colouring of a point set P that is not a good colouring of $G(P, \mathcal{R})$. The rectangle R shows that one of the edges of $G(P, \mathcal{R})$ is monochromatic.

Note that in the frequency assignment problem, the more stations in the range of a client with the same frequency, the more interference there is. Hence we want to bound the number of stations in the range of a client which have the same frequency. This motivates the study of k-conflict-free colourings, which were defined by Har-Peled and Smorodinsky [28]. We now introduce this concept, together with some related definitions.

Definition 3.1.5. Let P be a finite point set and $\mathcal{F}$ be a set of regions. Given a number $k \geq 1$, we define the following:

- A *k-conflict-free colouring* of $G(P, \mathcal{F})$ with m colours is a function $c : V \to [m]$ such that for each (hyper)edge e, there exists an $i \in [m]$ such that $|\{v \in e : \ c(v) = i\}| \leq k$.
- The *k-conflict-free colouring number* of $G(P, \mathcal{F})$, denoted by $cf_k(G(P, \mathcal{F}))$, is defined to be the minimum number of colours needed for a k-conflict-free colouring of $G(P, \mathcal{F})$.
- Given $n \in \mathbb{N}$ and a fixed set of regions $\mathcal{F}$, we define $cf_k(n, \mathcal{F})$ to be the minimum number of colours c_{min} so that we can find a k-conflict-free colouring of any point set of size n with c_{min} colours.

In the rest of this subsection we show a close relation between bounds on the chromatic number and conflict-free colouring number when the family $\mathcal{F}$ satisfies certain nice properties. This relation has been one of the main tools used so far to find upper bounds on the k-conflict-free colouring number. We will often abuse notation and refer to an edge (resp. vertex) by a region (resp. the point) corresponding to it. We begin by introducing a few parameters of $\mathcal{F}$, which will be our main focus of interest later in this chapter.

Definition 3.1.6. Given a set of regions $\mathcal{F}$ and a set of points P:

- Let $G_{\geq k}(P, \mathcal{F}) = (V, \mathcal{E})$, where $V = P$ and $E = \{f \cap P : \ f \in \mathcal{F}, \ |P \cap f| \geq k\}$.
- Let $G_k(P, \mathcal{F}) = (V, \mathcal{E})$, where $V = P$ and $E = \{f \cap P : \ f \in \mathcal{F}, \ |P \cap f| = k\}$. The graph $G_2(P, \mathcal{F})$ is often called the *Delaunay graph* of P with respect to $\mathcal{F}$.
- Let $\chi_k(\mathcal{F}) = \sup_{n \in \mathbb{N}} \chi_k(n, \mathcal{F})$.
- Define $\chi_k(n, \mathcal{F})$ to be the minimum number of colours c_{min} so that for all point sets with n elements we have $\chi(G_k(P, \mathcal{F})) \leq c_{min}$, where $\chi(G)$ is the chromatic number of G (see Definition 1.3.1). We will often denote $\chi_2(\mathcal{F})$ by $\chi(n, \mathcal{F})$.

Note that $cf(G(P, \mathcal{F})) = cf_1(G(P, \mathcal{F}))$ and $G(P, \mathcal{F}) = G_2(P, \mathcal{F})$. Also note that for all the families we have mentioned in this chapter, we have $cf_{k+1}(n, \mathcal{F}) \leq cf_k(n, \mathcal{F})$ (this is a consequence of Remark 3.1.9 below). We denote $\chi(G_k(P, \mathcal{F}))$ by $\chi_k(P, \mathcal{F})$.

Remark 3.1.7.

$$\chi_k(n, \mathcal{F}) = \max\{\chi(G_k(P, \mathcal{F})) : k \in \mathbb{N}\}.$$

Note that $\chi_k(\mathcal{F})$ may be unbounded, in which case, studying the asymptotic behaviour of $\chi_k(\mathcal{F})$ is often an interesting and challenging question. Also note that $\chi_k(G(P, \mathcal{F})) \geq 2$ when $G_k(P, \mathcal{F})$ has at least one edge.

Definition 3.1.8 (k-monotonic families)**.** Let P be a set of points and $\mathcal{F}$ be a set of regions. The ordered pair $(P, \mathcal{F})$ is called *k-monotonic* if for any region $r \in \mathcal{F}$ and point set $P' \subset P$ with $|r \cap P| > k$, there exists a region $r' \in \mathcal{F}$ with $|r' \cap P'| = k$ and $r' \cap P' \subset r \cap P'$. A family $\mathcal{F}$ is called *k-monotonic* if $(P, \mathcal{F})$ is k-monotonic for every point set P.

Keszegh [34] made the following observation, which had been implicitly stated by Har-Peled and Smorodinsky [28] beforehand.

Remark 3.1.9 (Keszegh [34])**.** *If $\mathcal{F}$ is k-monotonic, then $\chi(G_k(P, \mathcal{F})) = \chi(G_{\geq k}(P, \mathcal{F}))$ for every point set P.*

Proof. It is easy to see that $\chi(G_k(P, \mathcal{F})) \leq \chi(G_{\geq k}(P, \mathcal{F}))$ for any family $\mathcal{F}$ and any point set P. We now argue that $\chi(G_k(P, \mathcal{F})) \geq \chi(G_{\geq k}(P, \mathcal{F}))$. Given a good colouring c of $G_k(P, \mathcal{F})$, any edge e of size larger than k is a superset of an edge e' of size k, since $\mathcal{F}$ is k-monotonic. As we have a good colouring of $G_k(P, \mathcal{F})$, we have that e' is not monochromatic and hence e is not monochromatic. Therefore c is a good colouring of $G_{\geq k}(P, \mathcal{F})$. It follows that $\chi(G_k(P, \mathcal{F})) \geq \chi(G_{\geq k}(P, \mathcal{F}))$. □

The previous remark shows that for k-monotonic families, checking whether a colouring of a point set c is a good colouring of $G_{\geq k}(P, \mathcal{F})$ is equivalent to checking whether c is a good colouring of $G_k(P, \mathcal{F})$. All the families that we discuss in this thesis are k-monotonic for all k.

Remark 3.1.9 also implies that if a family $\mathcal{F}$ is k-monotonic, then $cf_{k+1}(n, \mathcal{F}) \leq cf_k(n, \mathcal{F})$.

Remark 3.1.10. *If $\mathcal{F}$ is a set of regions for which there exists a direction in which all the regions are bounded, then we have $\chi_k(G(n, \mathcal{F}) \leq \chi_k(G(n+1, \mathcal{F})$ for all $n, k \in \mathbb{N}$. We call such a family* bounded in one direction.

Proof (of Remark 3.1.10): Given a point set P, we want to add a point p to P so that $G_k(P, \mathcal{F})$ is a subgraph of $G_k(P \cup \{p\}, \mathcal{F})$. We choose this point as follows: Only finitely many regions are needed to induce all the edges of the graph $G_k(P, \mathcal{F})$, therefore we can always find a point p outside of all these regions in the bounded direction. By the choice of p, we have that $G_k(P, \mathcal{F})$ is a subgraph of $G_k(P \cup \{p\}, \mathcal{F})$. As a good colouring of a graph induces a good colouring of a subgraph, we are done. □

We include this criteria as it is sufficient for the sequence $(\chi_k(n, \mathcal{F}))_{n \in \mathbb{N}}$ to be monotonically increasing, i.e., it does not decrease.

This criteria is stronger than necessary, as can be seen in the case where $\mathcal{F}$ is the set of half planes in the plane. Even though this set is not bounded in one direction, the sequence $(\chi(n, \mathcal{F}))_{n \in \mathbb{N}}$ is monotically increasing. For our purposes this criteria is sufficient.

Even et al. [21] gave an algorithm that uses a bound on the chromatic number to bound the conflict-free colouring number. This has been applied in most of the results that bound $cf_k(n, \mathcal{F})$. Har-Peled and Smorodinsky [28] made a slight modification to this result, which was explicitly stated by Keszegh [34].

Lemma 3.1.11. *Let $n, k \in \mathbb{N}$ and let $\mathcal{F}$ be a k-monotonic family of regions that are all bounded in one direction. The following statements hold:*

1. *If $\chi_k(G(n, \mathcal{F})) \leq c$ for some constant $c \geq 2$, then $cf_{k-1}(G(n, \mathcal{F})) \leq \frac{\log(n)}{\log(\frac{c}{c-1})}$.*
2. *For $0 < \epsilon \leq 1$, if $\chi_k(G(n, \mathcal{F})) = O(n^\epsilon)$, then $cf_{k-1}(G(n, \mathcal{F})) = O(n^\epsilon)$.*

Proof of 1. Let $\chi_k(G(n,\mathcal{F})) \leq c$ for some constant c and consider an arbitrary point set P with n elements. We show that we can find a good k-conflict-free colouring c_f that uses at most $\frac{\log(n)}{\log(\frac{c}{c-1})}$ colours. We iteratively colour the points to construct c_f. Let $P_1 = P$. Given the point set P_i, take a good colouring c_i of $G(P_i, \mathcal{F})$ using c colours (This exists because of Remark 3.1.10). By the pidgeonhole principle, there exists a colour class X_i of c_i such that $|X_i| \geq \frac{|P_i|}{c}$. Let $c_f(p) = i$ for all points $p \in X_i$ and let $P_{i+1} = P_i \backslash X_i$. We repeat this until we get a set P_m with $|P_m| \leq k-1$. We then let $c_f(p) = m$ for all $p \in P_m$. By construction, c_f is a $(k-1)$-conflict-free colouring of $G(n,\mathcal{F})$. Indeed, let i be the last colour used to colour vertices in an edge e, then we must have $|X_i \cap e| \leq k-1$. If not, there must be a region $r' \in \mathcal{F}$ such that $|r' \cap X_i| = k$ because $\mathcal{F}$ is k-monotonic. Therefore, we have that $r' \cap X_i$ is an edge of $G_k(P_i, \mathcal{F})$, which has been coloured monochromatically by c_i. This contradicts that c_i is a good colouring of $G_k(P_i, \mathcal{F})$.
We now show that the number of colours used is at most $\frac{\log(n)}{\log(\frac{c}{c-1})}$. We get that $|P_1| = n$ and $|P_{i+1}| = |P_i| - \frac{|P_i|}{c}$. Solving the recursion gives $|P_m| = |P_1|(\frac{c-1}{c})^m$. If $m = \frac{\log(n)}{\log(\frac{c}{c-1})}$, then $|P_m| \leq k-1$. Therefore, the number of iterations needed is at most $\frac{\log(n)}{\log(\frac{c}{c-1})}$ and we are done.

Proof of 2. We use the same algorithm as in the proof of 1 to find a k-conflict-free colouring of $G(n,\mathcal{F})$. In each iteration we colour a set of vertices X_i of size $\frac{|X_i|}{C|X_i|^{\epsilon}}$ with colour i, where C is a constant. Once again, we obtain a recursion. The proof is completed by solving the recursion to show that the algorithm is terminated after $O(n^{\epsilon})$ iterations. $\square$

Let $\mathcal{H}$ be a family of homothetic copies of a bounded convex shape. By *homothetic copies* of a shape R, we mean a shape R' that can be obtained via a translation and a scaling of R. Pach and Tóth [47] proved that $cf(G(P,\mathcal{H})) = \Omega(\log(n))$ for every point set P of size n in the plane. Families of homothetic copies of a bounded convex shape R are 2-monotonic and bounded in one direction. A conflict-free colouring of a graph G is also a good colouring of G, therefore $\chi(G) \leq cf(G)$. Taking all these facts into account, Lemma 3.1.11 shows that understanding the asymptotic behaviour of $\chi_k(n,\mathcal{H})$ as n tends to infinity leads to an asymptotically close bound on $cf_k(n,\mathcal{H})$.

When arguing that c_f is a k-conflict-free colouring, we only use that the set of vertices X_i, which we colour in each iteration, is a k-independent set of $G_k(P_i,\mathcal{F})$. Here, a *k-independent set* of a graph G is defined to be a subset $\mathcal{I}_k$ of the vertices of G for which no edge of G is a subset of $\mathcal{I}_k$. Bounding the chromatic number from above is one method of showing the existance of a large independent set, but it is not necessary. Another thing to notice is that the algorithm above gives a k-conflict-free colouring with a special property. Namely, if e is an edge of $G(n,\mathcal{F})$, then $|\{p \in e : c(p) = m_e\}| \leq k$ where $m_e = \max_{p\in e}\{c(p)\}$, i.e., m_e the maximum index of a colour used in e. Equivalently, if we order the list of colours and let c_e be the highest colour used on vertices of an edge e, then there are at most k vertices $v \in e$ with $c(v) = c_e$. A colouring that satisfies this additional property is known as a *k-maximum colouring.* Given a graph G, the minimum number of colours that are needed for a k-maximum colouring is denoted by $\chi_{k\text{-}m}(G)$. It is easy to see that $cf_k(G) \leq \chi_{k\text{-}m}(G)$ for any graph G. In the case where the edges of G are all of size 2, we have $cf(G) = \chi_{k\text{-}m}(G)$. For $k > 2$, there exists graphs with n vertices, for which $cf(G) = 2$

and $\chi_{1\text{-}m}(G) = \Omega(n)$ (see [57]). Therefore, the difference between these values can be very big for graphs in general. Note that bounds on the k-conflict-free colouring number have usually been identified by constructing good k-maximum colourings as they are easier to handle.

Lemma 3.1.11 shows that computing k-maximum colourings does not seem too unreasonable, when bounding $cf(G(n,\mathcal{F}))$ for a k-monotonic family of regions $\mathcal{F}$ that is bounded in one direction; however, it is not always the best method to obtain tight upper bounds on the k-conflict-free colouring number for other graph classes that are not so well behaved. Smorodinsky [57] gives examples of graphs with $cf_3(G) = 2$ and $cf_{k\text{-}um}(G) = \lfloor |V|/2 \rfloor$.

3.1.3. Related Results

The study of conflict-free colourings of graphs was initiated by Even et al. [21]. Among other things, they investigated the conflict-free colouring number of range spaces $(G(P,\mathcal{F}))$ for various families of geometric objects in the plane such as discs in the plane ($\mathcal{D}$) and axis-aligned rectangles in the plane ($\mathcal{R}$). They found the algorithm, which is used in the proof of Lemma 3.1.11 to find conflict-free colourings. They used this algorithm, together with the fact that $G(P,\mathcal{D})$ is planar for any point set P, to show that $cf(n,\mathcal{D}) = O(\log(n))$, They showed that this $O(\log(n))$-colouring can be found in $O(n\log(n))$ time. They applied these ideas to obtain similar results when $\mathcal{F}$ is the set of half-planes in the plane. They also gave a lower bound on $cf(n,\mathcal{D})$. Namely, they show that for any set P of n points on a line, the graph $G(P,\mathcal{D})$ needs $\Omega(\log(n))$ colours for a good colouring. Pach and Tóth [47] proved the following statement: $cf(G(P,\mathcal{F})) \geq \log_8(n)$ for every point set P of size n in the plane and any family $\mathcal{F}$ of homothetic copies of given convex shape.

Regarding axis-aligned rectangles, Har-Peled and Smorodinsky [28] gave a proof that applies the Erdös-Szekeres Lemma about the existence of monotone subsequences to prove that $cf(n,\mathcal{R}) = O(\sqrt{n})$. Pach and Tóth [47] improved this bound slightly and showed that $cf(n,\mathcal{R}) = O\left(\sqrt{\frac{n\log\log(n)}{\log(n)}}\right)$. The best known upper bound is by Ajwani et al. [1], who show $cf(n,\mathcal{R}) = \tilde{O}(n^{0.382})$, where $\tilde{O}$ includes a polylogarithmic factor. The best known lower bound is $\Omega(\log(n))$, which is achieved by any set of n- points on a line. A large gap remains between the lower bound and the upper bound of $cf(n,\mathcal{R})$, which seems much more challenging to close than in the case of discs.

Trying to improve the bounds known for axis-aligned rectangles has motivated trying to show the existance of a large independent set in $G(P,\mathcal{R})$ for any point set P. Using probabalistic arguments, Chen et al. [14] show that there does not exist an $\Omega(n)$ lower bound on the independence number for point sets of size n. In fact, they show that the graph $G(P,\mathcal{R})$ of a randomly and uniformly selected set of n points in the unit square almost surely has an independence number of size $O\left(n\frac{\log^2\log(n)}{\log(n)}\right)$. Applying the fact that $\chi(G) \geq \frac{n}{\alpha(G)}$, we get the existence of sets P of n points in the unit square $G(P,\mathcal{R})$ whose chromatic number is of size $\Omega\left(\frac{\log(n)}{\log^2\log(n)}\right)$. They also generalised this result to hypergraphs, showing that the k-independence number of the graph $G_k(P,\mathcal{R})$ of a randomly and uniformly selected set of points in the unit square is almost surely of size $O\left(kn\frac{\log^2\log(n)}{\log^{1/(k-1)}(n)}\right)$. Here, the *$k$-independence number* is the size of the biggest k-independent set. It follows that for all values of k, the value of $\chi_k(n,\mathcal{F})$ tends to infinity as n tends to infinity. They also apply

a theorem by Spencer [58] to show that every hypergraph $G(P, \mathcal{R})$ of a point set P of size n has an k-independent set of size $\Omega(n/(\log^{1/(k-1)}(n))))$.
In the case when $k = 2$ the best known lower bound is by Ajwani et al. [1], they show there exists an independent set of size $\tilde{\Omega}(n^{0.382})$.

For some families $\mathcal{F}$ where $\chi_k(n, \mathcal{F})$ is bounded by a constant, there has also been interest in computing the value of $\chi_k(\mathcal{F})$. Recall that $\chi_k(\mathcal{F})$ is the least upper bound of $\chi_k(P, \mathcal{F})$ over all finite point sets P. Keszegh [34] found the values of $\chi_k(\mathcal{F})$ for all $k \in \mathbb{N}$ when $\mathcal{F}$ is the set of all bottomless rectangles (see Section 3.2). He also found $\chi_k(\mathcal{F})$ in the case where $\mathcal{F}$ is the set of all half-planes in the plane. Regarding the family of discs in the plane, $\chi_2(G(P, \mathcal{D}))$ is planar and hence 4-colourable. From the monotonicity of $\mathcal{D}$ it follows that $\chi_k(\mathcal{D}) \leq 4$ for all k. Pach et al. [46] gave a construction of a family of point sets P_k such that $\chi_k(P_k, \mathcal{D}) = 3$. It follows that $\chi_k(\mathcal{D}) \geq 3$ for all $k \in \mathbb{N}$. It remains open whether there exists a natural number k for which $\chi_k(\mathcal{D}) = 4$.

Regarding the k-conflict-free colouring number for $k \geq 2$, Har-Peled and Smorodinsky [28] showed that there are families for which $cf_k(G(n, \mathcal{F}))$ does not have the same value for all k. More specifically, they showed that $cf_k(G(n, \mathcal{B}_2)) = O(n^{1/k})$ for any $k \in \mathbb{N}$, where $\mathcal{B}_2$ denotes the set of all balls in $\mathbb{R}^3$. They also give an example of a set P of n points in $\mathbb{R}^3$ for which $cf_k(G(P, \mathcal{B}_2)) = n$. This implies that $cf_k(n, \mathcal{F})$ and $cf_{k+1}(n, \mathcal{F})$ are not the same in general.

In addition to the problems mentioned above, other colouring problems of this nature that can be found in the literature include:

- Conflict-free colourings of regions (see the discussion after Definition 3.1.2). Colourings of regions are closely related to the notion of decomposable coverings of the plane (see Pach et al. [46]).
- Online variations of colouring problems on point sets, i.e., the points are introduced one after another and a point must be coloured immediately after introduction.
- The unique-maximum colouring number and how it relates to the conflict-free colouring number (for graphs in general, and for some specific graph classes).
- Finding efficient approximation algorithms for the conflict-free colouring number. This is motivated by the fact that computing this number is hard in general, even where $\mathcal{F}$ is the set of all discs in the plane with radius 1 (see Even et al. [21]).

For a more detailed overview of conflict-free colourings and the results known to date, we refer the reader to [57].

3.2. Axis-Aligned Rectangles that are Pierced by a Line

In this section, we investigate bounds on $\chi_k(\mathcal{F})$ for a specific subfamily of axis-aligned rectangles. Given a fixed horizontal line L and $k \in \mathbb{N}$, we let $\mathcal{R}_L$ be the set of axis-aligned rectangles that intersect L (see Figure 3.6). Keszegh [34] gave some bounds on $\chi_k(\mathcal{R}_L)$ for all values of k. We find new lower bounds on $\chi_k(\mathcal{F})$ for some values of k, leaving $k = 6$ as the only unsolved case.

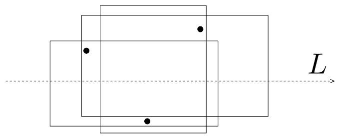


Figure 3.6.: Three Rectangles that are in $\mathcal{R}_L$, together with a point set P.

Note that in Figure 3.6 $G_2(P, \mathcal{R}_L)$ is a 3-cycle, but if we would translate the entire pointset above L, then the graph would be a path of length 2 (the same graph that we would obtain if the regions were bottomless rectangles).

One motivation for studying the family $\mathcal{R}_L$ is as follows: $\mathcal{R}_L$ can be generalised to the family $\mathcal{R}_{L_1,L_2,\ldots,L_m}$ of rectangles that intersect at least one of the horizontal lines $L_1, L_2, .., L_m$. If we have a point set $P \subset \mathbb{R}^2$, we can find a number m_P that is big enough so that $G(P, \mathcal{R}) = G(P, \mathcal{R}_{L_1,L_2,..,L_{m_P}})$. Therefore, understanding the structure of $G(P, \mathcal{R}_{L_1,L_2,..,L_m})$ as m increases would lead to more insight in the structure of $G(P, \mathcal{R})$.

Keszegh [34] used the values of $\chi_k(\mathcal{B})$, together with Lemma 3.2.2 and Lemma 3.2.3 to give the upper bounds on $\chi_k(\mathcal{R}_L)$. These bounds are given in Table 3.1. We give examples of point sets showing that these upper bounds on $\chi_k(\mathcal{R}_L)$ are tight for $k = 1, \ldots, 5$. It follows that the only unknown value of $\chi_k(\mathcal{R}_L)$ is when $k = 6$, whose value is either 2 or 3.

Table 3.1.: Bounds on $\chi(\mathcal{R}_L)$ shown by Keszegh [34].

$k = 2$	$3 \leq k \leq 6$	$k \geq 7$
$4 \leq \chi_k(\mathcal{R}_L) \leq 6$	$2 \leq \chi_k(\mathcal{R}_L) \leq 3$	$\chi_k(\mathcal{R}_L) = 2$

In this chapter all point sets are presumed to be in general position. Here, a point set is said to be in *general position* if no two points have the same x- or y-coordinates and no point lies on the horizontal line L (in the case when looking at $\mathcal{R}_L$). Note that because we are looking at finite sets of points, for any point set P that is not in general position we can perturb the points slightly to obtain a set of points P' in general position such that edges of $G(P, \mathcal{R}_L)$ and $G(P, \mathcal{B})$ are also edges of $G(P', \mathcal{R}_L)$ and $G(P', \mathcal{B})$, respectively. This implies that $\chi_k(G(P', \mathcal{R}_L)) \geq \chi_k(G(P, \mathcal{R}_L))$ and $\chi_k(G(P', \mathcal{B})) \geq \chi_k(G(P, \mathcal{B}))$. Therefore, restricting ourselves to point sets in general position does not change the values of $\chi_k(\mathcal{R}_L)$ and $\chi_k(\mathcal{B})$ that we obtain. In the remainder of this subsection we show some relations between the families $\mathcal{R}_L$ and $\mathcal{B}$.

Remark 3.2.1. *Given a finite point set P, let L_1 (resp. L_2) be a horizontal line below (resp. above) all points in P. The following holds:*

$$\chi_k(G(P, \mathcal{R}_{L_1})) \leq \chi_k(\mathcal{B}) \text{ and } \chi_k(G(P, \mathcal{R}_{L_2})) \leq \chi_k(G(\mathcal{B})).$$

Proof. The inequality on the left holds because every bottomless rectangle can be seen as a rectangle that intersects a line below all of the points in P. The statement on the right holds because we could reflect the point set in line L_2 to obtain a new point set P' above L_2. Edges that bottomless rectangles induce on P' then correspond to edges that rectangles in $\mathcal{R}_{L_2}$ induce on P. Therefore $\chi_k(G(P, \mathcal{R}_{L_2})) = \chi_k(G(P', \mathcal{B})) \leq \chi_k(\mathcal{B})$. $\square$

Using Remark 3.2.1 we get the following two lemmas, which are the key tools used by Keszegh [34] for obtaining upper bounds on $\chi_k(\mathcal{R}_L)$.

Lemma 3.2.2. *Given a point set P, then*

$$\chi_k(G(P,\mathcal{R}_L)) \leq 2\chi_k(\mathcal{B}) \text{ for all } k \in \mathbb{N}.$$

Proof. Given a line L and a point set P, we partition P into the sets P_{upper} and P_{lower}, where P_{upper} is the set of points in P that lie above L and P_{lower} is the set of points in P that below the L. This is indeed a partition due to the points being in general position. By Remark 3.2.1, we can find a colouring of both P_{upper} and P_{lower} using $\chi_k(\mathcal{B})$ colours so that there is no monochromatic edge in $G(P_{upper},\mathcal{R}_L)$ or $G(P_{lower},\mathcal{R}_L)$. We colour P_{upper} and P_{lower} using disjoint sets of colours. We claim that this gives a good colouring of $G_k(P,\mathcal{R}_L)$. Consider an rectangle $R_e \in \mathcal{R}_L$ that induces an edge e of $G(P,\mathcal{R}_L)$. Either R_e contains k points that all lie on the same side of the line L, or R_e contains points on both side of L. In the first case, e cannot be monochromatic as it is an edge of $G(P_{upper},\mathcal{R}_{\mathcal{L}})$ or $G(P_{lower},\mathcal{R}_{\mathcal{L}})$. In the second case, e cannot be monochromatic as we have used disjoint sets of colours for P_{lower} and P_{upper}. □

Lemma 3.2.3. *Given a point set P, then*

$$\chi_{2k-1}(G(P,\mathcal{R}_L)) \leq \chi_k(\mathcal{B}) \text{ for all } k \in \mathbb{N}.$$

Proof. Similarly to the proof of the previous lemma, we colour P_{upper} and P_{lower} separately with a good $\chi_k(\mathcal{B})$-colouring. This time however, we use the same set of colours to colour P_{upper} and P_{lower}. Let R be a rectangle that induces an edge e of $G_{2k-1}(P,\mathcal{R}_L)$. By definition, R must contain $2k-1$ points in P. The pidgeonhole principle implies that R must contain at least k points that lie on one side of L. Therefore R must induce an edge e' in either $G(P_{upper},\mathcal{R}_L)$ or $G(P_{lower},\mathcal{R}_L)$. By the construction of the colouring, e' is not monochromatic, and therefore e is not monochromatic. Hence, we have a good colouring of $G_{2k-1}(P,\mathcal{R}_L)$ with $\chi_k(\mathcal{B})$ colours.

□

3.2.1. Bounding $\chi_2(\mathcal{R}_L)$ from Below

In this section we give an example of a point set P such that $\chi_2(P,\mathcal{R}_L) = 6$, from which it follows that $\chi_2(\mathcal{R}_L) = 6$. We begin by presenting the upper bound $\chi_2(\mathcal{R}_L)$, which was found by Keszegh [34]. For this we need to look at bottomless rectangles. The following claim, which Keszegh [34] calls 'folklore', shows that $\chi_2(\mathcal{B}) \leq 3$.

Claim 3.2.4. $\chi_2(\mathcal{B}) \leq 3$.

Proof. Given a point set P, we find a 3 colouring of $G(P,\mathcal{B})$. We do this by colouring the points in increasing order with respect to their y-coordinate, maintaining the following invariant.

Invariant: Let P_m be the set of points that we have already coloured.
If $p_1,..,p_m$ is the ordering of the points in P_m with respect to their x-coordinates, such that subsequent points are not of the same colour, i.e., $c(p_i) \neq c(p_{i+1})$ for $1 \leq i \leq m-1$.

We can maintain the invariant because when we colour a point, we have three colours to choose from. There are only at most two points in the ordering with respect to their x-coordinates whose colour we may not use. Therefore there is always one free colour. It remains to show that this is a good colouring of G. Let e be an edge of G, and let p be the last point to be coloured in e. If p is the mth point that we colour, and i is its index in the ordering $p_1, .., p_m$, then the other point in e must be one of p_{i+1} or p_{i-1}. Therefore, by the induction hypothesis, e is not monochromatic. □

The next corollary follows from Lemma 3.2.2.

Corollary 3.2.5 (Keszegh [34])**.** $\chi(\mathcal{R}_L) \leq 6$.

Note that $\chi_2(\mathcal{B}) = 3$, as one can find a point set P such that $\chi_k(P, \mathcal{B}) = 3$ (see Figure 3.7).

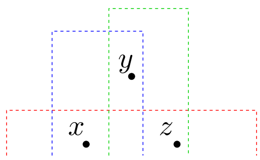


Figure 3.7.: A point set for which $\chi_2(G(P, \mathcal{R}_L)) = 3$.

We will now show that $\chi_2(\mathcal{R}_L) = 6$ with the following proposition.

Proposition 3.2.6. *For the point set P in Figure 3.8 we have $\chi_2(G(P, \mathcal{R}_L)) = 6$.*

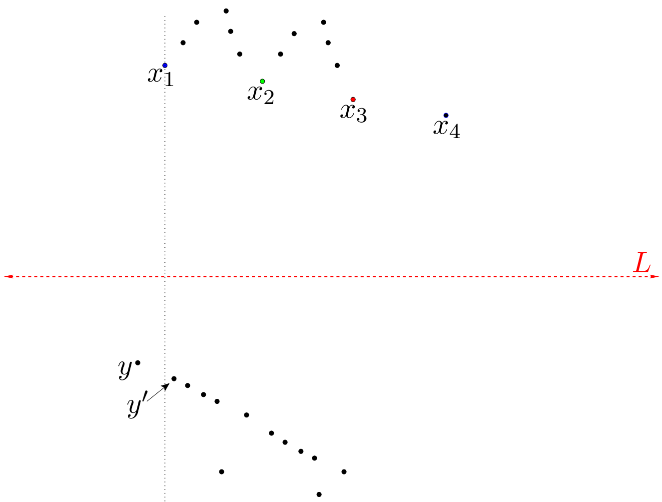


Figure 3.8.: A point set P for which $\chi_2(\mathcal{R}_L) = 6$.

Proof. We assume there exists a good colouring c of $G(P, \mathcal{R}_L)$ with 5 colours. This will lead to a contradiction. First consider the points in $\{x_1, x_2, x_3, x_4, y, y'\}$ in Figure 3.8.
Fact 1. There exists $j \in \{1, 2\}$ and $k \in \{3, 4\}$ such that $c(x_j) = c(x_k)$.
If not, then as (x_1, x_2) and (x_3, x_4) are both edges, we must have that all the x_i's have different colours. However, as y and y' are both adjacent to all x_i they must both receive

the 5th colour. Because (y, y') is an edge, we get a contradiction and hence, fact 1 must hold. Now that such a pair x_j, x_k exists, we can assume without loss of generality that $c(x_j) = c(x_k) = 1$. Also assume without loss of generality that $c(x_{j+1}) = 2$. Consider the point set restricted to the vertical strip between x_j and x_{j+1}. The relative positions of the vertices in the strip are indicated more clearly in Figure 3.9.

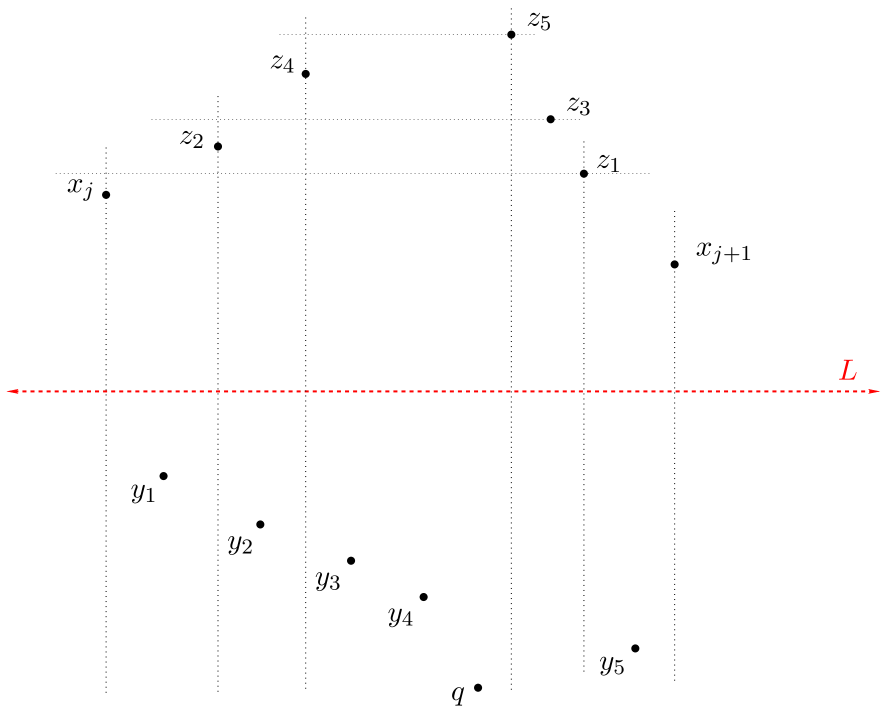


Figure 3.9.: A closer look at the section between x_j and x_{j+1}.

Given the vertices labeled as in Figure 3.9, we now show the following fact.

Fact 2. $c(y_i) \notin \{1, 2\}$ for all $i = 1, \ldots, 5$.

Consider a point y_i for $i \in \{1, \ldots, 5\}$. As y_i is adjacent to x_{j+1}, we have that $c(y_i) \neq 2$. Additionally, because of how we chose x_j, the vertex x_k is adjacent to y_i and $c(x_k) = c(x_j) = 1$. Therefore $c(y_i) \neq 1$.

Fact 3. We can use at most 2 different colours on the set $\{y_1, .., y_5\}$.

The vertex z_1 is adjacent to vertices x_j and x_{j+1}, therefore $c(z_1) \notin \{1, 2\}$. Without loss of generality, let $c(z_1) = 5$. Now, for $i = 1, \ldots, 5$ we have that z_1 is adjacent to y_i. Therefore, together with Fact 2, we have that $c(y_i) \in \{3, 4\}$ for $i = 1, \ldots, 5$.

Fact 4. $c(z_1) = 5, c(z_3) = 1, c(z_5) = 2$:

For all i, there exists a number m such that z_i is adjacent to y_m and y_{m+1}. Therefore, by Fact 3, together with the fact that $\{y_m, y_{m+1}\} \in \mathcal{E}$, we have $c(z_i) \in \{1, 2, 5\}$ for all i. Now, consider the induced subgraph on the vertex set $\{x_j, x_{j+1}, z_1, z_2, z_3, z_4, z_5\}$ (see Figure 3.10). The vertices x_j and x_{j+1} already have colours 1 and 2, which forces the colour of the z_i for all i. The statement in Fact 4 follows.

Now vertex q is adjacent to y_4, y_5, z_1, z_3, z_5, which have received all five of the colours available (by Facts 3 and 4). Therefore, there is no colour available for q, which implies that there does not exist a good colouring of $G(P, \mathcal{R}_L)$ with 5 colours.

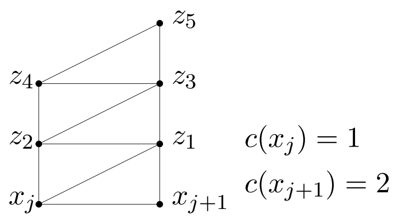


Figure 3.10.: Restriction on $\{x_j, x_{j+1}, z_1, z_2, z_3, z_4, z_5\}$.

By Corollary 3.2.5 we get $\chi(G(P, \mathcal{R}_L)) = 6$. □

We have shown that $\chi_2(\mathcal{R}_L) \geq 6$. Therefore, from the upper bound proved by Keszegh [34], we get the following result:

Theorem 3.2.7. $\chi_2(\mathcal{R}_L) = 6$.

3.2.2. $\chi_k(R_L)$ for $k > 2$

In this subsection we investigate $\chi_k(\mathcal{R}_L)$ for $k \geq 3$. First note that as the family $\mathcal{B}$ of bottomless rectangles is monotonic, we get that $\chi_k(\mathcal{B}) \leq \chi_2(\mathcal{B})$ by Remark 3.1.9. By Claim 3.2.4, we know that $\chi_2(\mathcal{B}) \leq 3$. Therefore we have $\chi_k(\mathcal{B}) \leq 3$ for all $k \geq 2$. Applying Lemma 3.2.3, we get that $\chi_3(\mathcal{R}_L) \leq 3$. We will show that $\chi_5(\mathcal{R}_L) \geq 3$ and therefore

$$3 \leq \chi_5(\mathcal{R}_L) \leq \chi_4(\mathcal{R}_L) \leq \chi_3(\mathcal{R}_L) \leq 3,$$

which implies $\chi_k(\mathcal{R}_L) = 3$ for $k = 3, 4, 5$.

Proposition 3.2.8. $\chi_5(\mathcal{R}_L) \geq 3$.

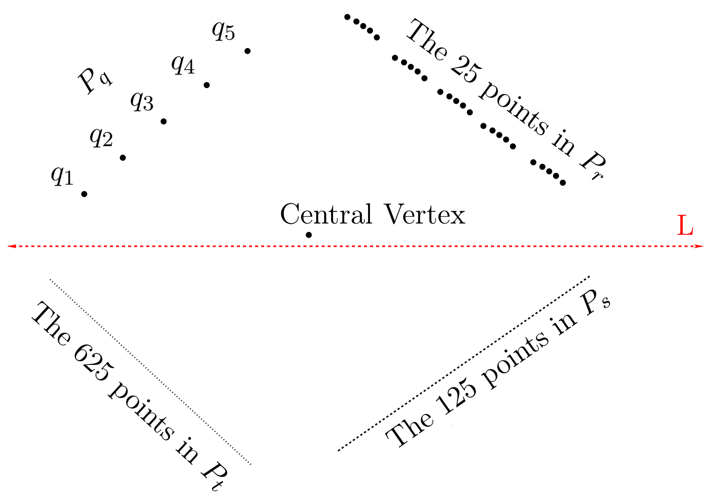


Figure 3.11.: A point set P for which $\chi_5(G(P, \mathcal{R}_L)) \geq 3$.

Proof. We give a point set P for which $G_5(P, \mathcal{R}_L)$ is not 2-colourable. An idea of the overall structure of the point set P can be seen in Figure 3.11. The point set consists of a central point together with 780 other points, which we partition into four sets, $P_q = \{q_i : i \leq 5\}$,

$P_r = \{r_i : i \leq 25\}$, $P_s = \{s_i : i \leq 125\}$, and $P_t = \{t_i : i \leq 625\}$ (indicated in Figure 3.11). The sets P_q and P_r lie above the line L and the sets P_s and P_t lie below the line L. All the points in the set P_q and P_t lie completely to the left of all the points in P_r and P_s.

The figures below indicate the relative positions of the points in more detail. The range of i for which they are true is indicated in the captions of the figures. Note that the figures have been scaled so that it is easier to see the relative positions.

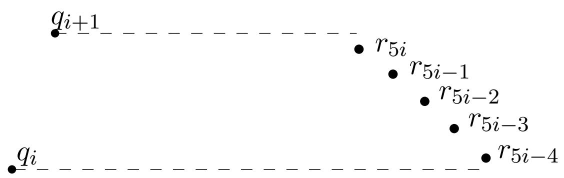


Figure 3.12.: The relative positions for $1 \leq i \leq 4$

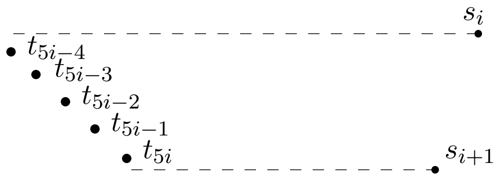


Figure 3.13.: The relative positions for $1 \leq i \leq 124$.

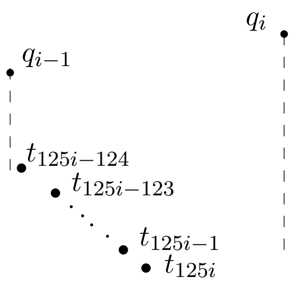


Figure 3.14.: The relative positions for $2 \leq i \leq 5$.

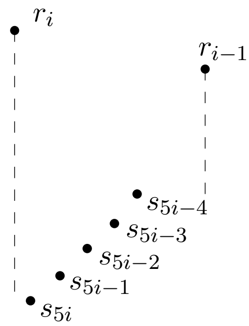


Figure 3.15.: The relative positions for $2 \leq i \leq 25$.

A closer look at the relative positions of the points.

- Removing the point q_{i+1} from Figure 3.12 and letting $i = 5$, shows the relative positions of $q_5, r_{21}, r_{22}, r_{23}, r_{24}, r_{25}$.
- Removing the point s_{i+1} from Figure 3.13 and letting $i = 125$, shows the relative positions of $t_{625}, t_{624}, t_{623}, t_{622}, t_{621}, s_{125}$.
- Removing the point q_{i-1} from Figure 3.14 and letting $i = 1$, shows the relative positions of $r_1, s_5, s_4, s_3, s_2, s_1$.
- Removing the point r_{i-1} from Figure 3.15 and letting $i = 1$, shows the relative positions of $t_1, t_2, \ldots, t_{124}, t_{125}, q_1$.

These figures show a labeling of the points, which we will use in the remainder of the proof. Assume that we have a good colouring c of $G_5(P, \mathcal{R}_L)$ with two colours (1 and 2). We will show that c must have a monochromatic edge, i.e., there is a rectangle that contains 5 points that are all coloured with the same colour. Suppose that we have a good colouring of P with 2 colours. Without loss of generality, the central point is coloured 1. The set P_q is an

edge of $G_5(P, \mathcal{R}_L)$. Therefore, as we have a good colouring of P, not all of the points in P_q can receive the colour 2. Let q_i be the point in P_q that is coloured 1.
Arguing similarly, one of the points $r_{5i-4}, .., r_{5i}$ that lie between q_i and q_{i+1} also receives colour 1, call this r_j. In the same manner, one of $s_{5j-4}, .., s_{5j}$ also receives colour 1, call this s_k. Finally, we must also have that one of $t_{5k-4}, .., t_{5k}$ receives colour 1, call this z_l. By construction of the point set, the four points q_i, r_j, s_k, t_l together with the central point p_c are an edge of G.

Indeed, we let the bottom left corner of a rectangle R in $\mathcal{R}_L$ be t_l. Choose the rectangle R so that s_k lies on the right boundary of R. One can find such a rectangle in $\mathcal{R}_L$ as s_k and t_l both lie below L, and the y-coordinate of t_l is less than the y-coordinate of s_k. We chose R so that r_j lies on the top boundary of R. By definition of the point set, such a rectangle exists and also contains q_i. By the relative positions of the points, R only contains the points p_c, q_i, r_j, s_k, t_l. Therefore R induces an edge of $G(P, \mathcal{R}_L)$ that has been coloured monochromatically. An example of such a rectangle R is shown in Figure 3.16. $\square$

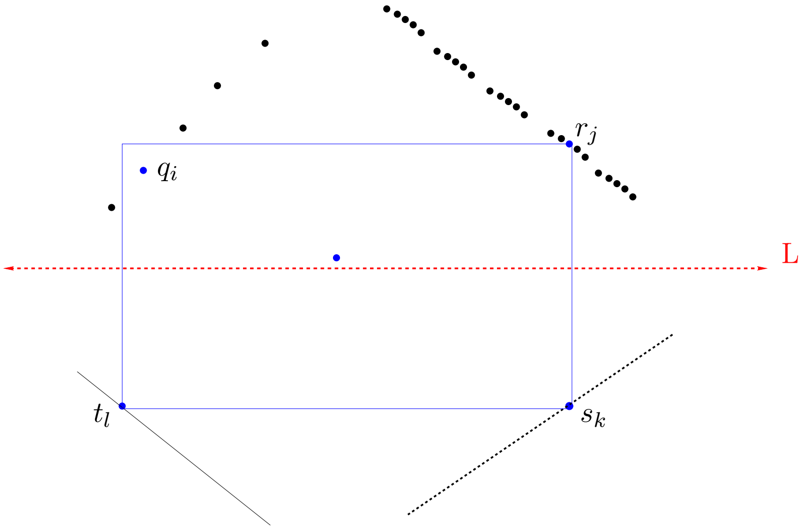


Figure 3.16.: A monochromatic rectangle.

Corollary 3.2.9. $\chi_k(\mathcal{R}_L) = 3$ *for* $3 \leq k \leq 5$.

Keszegh shows that $\chi_k(\mathcal{B}) = 2$ for $k \geq 4$, for the bottomless rectangles. He then applies Lemma 3.2.3 to show that $\chi_k(\mathcal{R}_L) \leq 2$ for $k \geq 7$. We include this result for completeness.

Proposition 3.2.10 (Keszegh [34])**.** *Given any point set P, we can find a good colouring c of $G_4(P, \mathcal{B})$ with 2 colours.*

Proof. We process the points in increasing order with respect to their y-coordinate. By *processing a point p*, we mean that we either colour p and possibly another uncoloured point in P, or we leave p uncoloured so that it can be coloured later. We process the points so that the following invariant is maintained:

Invariant: Let P_k be the set of k points that we have already processed. If $p_1, p_2, \ldots, p_k$ is the ordering of the points in P_k with respect to their x-coordinates, then we have:

1. No two consecutive points in this ordering are uncoloured.
2. If two points p_i and p_j with $i < j$ have been coloured with the same colour, then there exists another point p_l in P_k with $i < l < j$ that has been coloured with the other colour.

We maintain the invariant by colouring as follows. We begin by colouring the first vertex with one of the colours, this clearly satisfies the invariant. Now, suppose the invariant has been maintained when processing the first k points. Let p be the next point we need to process and let $p_1, p_2, \ldots, p_{k+1}$ be the ordering of the points in $P_k \cup \{p\}$ with respect to their x-coordinates. Let m be the index of p in the ordering $p_1, p_2, \ldots, p_{k+1}$, i.e., $p = p_m$. If neither of the points p_{m-1} and p_{m+1} is uncoloured, then we leave p_m uncoloured and it is easy to see that we have maintained the invariant. Otherwise, there is a point $p' \in \{p_{m-1}, p_{m+1}\}$ which is uncoloured. Without loss of generality $p' = p_{m+1}$.

If $m \notin \{1, k\}$, then p_{m-1} and p_{m+2} have both been coloured already because the first part of the invariant holds for P_k. By the second part of the invariant $c(p_{m-1}) \neq c(p_{m+2})$. We let $c(p_{m+1}) = c(p_{m-1})$ and $c(p_m) = c(p_{m+2})$. When $m = 1$, then p_3 must have been coloured. We let $c(p_1) = c(p_3)$ and we let $c(p_2) \neq c(p_1)$. The case when $m = k$ is similar. Colouring in this way, we have ensured that the invariant holds for P_{k+1}.

Once we have processed all the points, we colour the remaining uncoloured points arbitrarily. We now show that a colouring c obtained in this way gives a good colouring of $G_4(P, \mathcal{B})$. Let B be a bottomless rectangle that contains exactly 4 contains in P. Consider the last point p that is processed in B. suppose p is the kth point that is processed. By the definition of a bottomless rectangle, we have that the four points in B are consecutive in the ordering $p_1, p_2, \ldots, p_k$ of P_k. From the first part of the invariant on P_k, we must have at least 2 points in B that have been coloured before the $(k+1)$th point is processed. The second part of the invariant on P_k implies at least 2 of the points in B must have been coloured with different colours once p is processed. Therefore, B does not induce a monochromatic edge. This shows that c is a good colouring of $G_4(P, \mathcal{B})$ and we are done. □

By Lemma 3.2.3, together with the fact that $\mathcal{R}_L$ is k-monotonic, we can conclude the following:

Corollary 3.2.11 (Keszegh [34]). *$\chi_k(\mathcal{R}_L) = 2$ for all $k \geq 7$.*

The only unknown value remaining is $\chi_6(\mathcal{R}_L)$, which is either 2 or 3 because

$$3 = \chi_5(\mathcal{R}_L) \geq \chi_6(\mathcal{R}_L) \geq 2.$$

4. Summary

As the problems that we consider in Chapter 2 and Chapter 3 are quite different, we give a seperate conclusion for each chapter.

4.1. Summary and Open Questions for Chapter 2.

We have investigated cyclic segment graphs and hook graphs and we have given various models of them. Figure 4.1 shows relations between hook graphs, cyclic segment graphs and some other known graph classes, where an edge between two classes implies that the highest of the two classes contains the other.

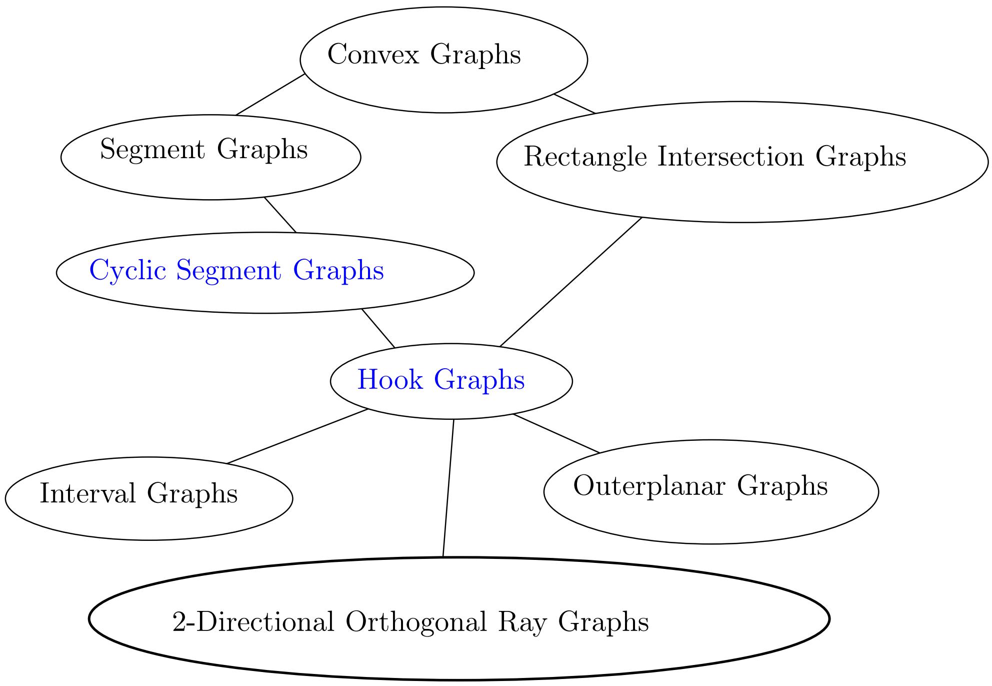


Figure 4.1.: Subclasses and superclasses

We have proved that the recognition problem for cyclic segment graphs is NP-complete. Whilst proving the NP-completeness result, we proved that bipartite cyclic segment graphs are exactly grid intersection graphs. Bellantoni et al. [5] have proved that grid intersection graphs are exactly bipartite rectangle intersection graphs. Therefore a bipartite graph is a cyclic segment graph if and only if it is a rectangle intersection graph. Cantazaro et al. [9] have shown that computing the chromatic number of hook graphs is NP-hard, it follows that computing the chromatic number of cyclic segment graphs is NP-hard. The complexity of computing the clique number, the independence number, and the clique covering remains unknown for cyclic segment graphs. We have shown that the clique number and the independence number can be computed in polynomial times for hook graphs. It has also been shown by Cabello et al. [8] (resp. J. Chalopin and D. Gonçalves [13]) that the clique number (resp. independence number) is NP-hard to compute for segment graphs. It would be interesting to see whether a polynomial time algorithm that computes any of these values for cyclic segment graphs exists. Another interesting way of extending ideas would be to consider another set of lines that is not the cyclic arrangement, i.e., for which there does not exist a homeomorphism of the plane that maps them to a set of lines that are tangent to a parabola. We also mentioned that the graph $K_{2,2,2}$ has a segment representation whose segments lie on lines that are tangent to a circle, but we have shown that $K_{2,2,2}$ is not a hook graph. It would also be interesting to investigate graphs that have segment representations whose segments lie on lines that are tangent to the circle, and compare this class to hook graphs.

Regarding Hook graphs, one of the most fascinating problems is to find a bound on the chromatic number in terms of the clique number that is asymptotically best possible. This is particularly interesting as it is widely thought that one can bound the chromatic number rectangle intersection graphs in general by a function that is linear in the clique number. We have tried to extend the ideas used when showing that triangle-free hook graphs are 4-colourable, but have not been successful. We have also shown that all odd holes do not have a hook representation except for C_5, which is a cycle. We have also shown that the hook representations of cycles is quite limited and that we can always find three vertices v, w, and x in a odd hole C_{2k+1} with $v <_h w <_h x$ such that (v, x) is an edge of C_{2k+1}, and (v, w) and (w, x) are not edges of C_{2k+1}. Removing such a vertex w from each odd hole in a hook graph would leave us with a perfect graph; however, we have not found an effective way of doing this to find a better bound on the chromatic number. Another problem that would be very interesting is to compute the complexity of the recognition problem for hook graphs. We have shown that the recognition problem for cyclic segment graphs is NP-complete. It also follows from the fact that bipartite rectangle intersection graphs are grid intersection graphs that the recognition problem for rectangle intersection graphs is also NP-complete. However, for all the classes of graphs for which we have found hook representations, the recognition problem can be solved in polynomial time. In Theorem 2.1.10, we showed that the recognition problem for bipartite cyclic segment graphs is NP-complete, However, the complexity of the recognition problem remains unknown, even for bipartite hook graphs and stick graphs. We have shown that 2-directional orthogonal ray graphs are stick graphs. It is also known that one can recognise them 2-directional orthogonal ray graphs in polynomial time as their complements are circular arc graphs, which can be recognised in polynomial time (see McConnell [42]). Stefan Felsner (personal communication) has noticed that stick graphs are comparability graphs of 3-dimensional posets of height 2. Veit Wiechert (personal communication) has generalised this result by showing that bipartite hook graphs are also comparability graphs of 3-dimensional posets of height 2. Yannakakis [61] has showed that testing whether a height 2 poset has dimension at most k, is NP-complete for $k \geq 4$. Testing whether a poset is of dimension at most 2 can be done in polynomial time (see Golumbic [27]). It remains unknown, whether testing if a poset of height 2 has dimension at most 3 is NP-complete. This gives much motivation for investigating the complexity of the recognition problem for stick graphs and bipartite hook graphs.

4.2. Summary and Open Questions for Chapter 3.

We have shown that the bound of 6 on the chromatic number of graphs $G(P, \mathcal{R}_l)$ is best possible, i.e., we have given a point set P such that the chromatic number of $G(P, \mathcal{R}_l)$ is 6. We have also shown that the upper bound of 3 on graphs $G_k(P, \mathcal{R}_l)$ is best possible for $3 \leq k \leq 6$. It remains unknown whether the upper bound of 3 is best possible for graphs $G_6(P, \mathcal{R}_l)$. An example of a point set P such that the graph chromatic number of the graph $G_6(P, \mathcal{R}_l)$ remains unknown. It is also unknown whether we can in fact find a good 2-colouring of $G_6(P, \mathcal{R}_l)$ for any point set in the plane. Solving this question would give tight bounds for all values of k as it is known that $G_k(P, \mathcal{R}_l)$ is 2-colourable for any $k \geq 7$. One could also generalise these questions to the set $\mathcal{R}_{l_1, l_2, \ldots, l_n}$ of all rectangles that intersect at least one of n fixed horizontal lines $l_1, l_2, \ldots, l_n$. This problem seems more complicated as there is no simple way of using the bounds on bottomless rectangles to obtain good bounds on $G(P, \mathcal{R}_{l_1, l_2, \ldots, l_n})$.

A. Deutsche Zusammenfassung

In dieser Arbeit konzentrieren wir uns auf einige Probleme in der geometrischen Graphentheorie. Es gibt zwei Hauptthemen, die wir untersuchen. In dem Hauptteil (Kapitel 2) dieser Masterarbeit beschäftigen wir uns mit Schnittgraphen von geometrischen Objekten in der Ebene. Die Arbeit in Kapitel 2 wurde ursprünglich durch die Untersuchung von Segmentgraphen motiviert. Ein Graph ist ein *Segmentgraph*, wenn der Graph durch Segmente von Geraden in der Ebene dargestellt werden kann. Dabei ist (v, w) eine Kante genau dann, wenn sich die zwei entsprechenden Segmente schneiden. Wir führen eine Klasse von Segmentgraphen ein, ihre Elemente nennen wir zyklische Segmentgraphen. Hierbei ist ein Graph ein zyklischer Segmentgraph, wenn es eine Segmentdarstellung hat, dessen Segmente alle auf Tangenten einer Parabel liegen und keine zwei Segmente parallel sind. Wir präsentieren verschiedene Modelle dieser Graphen und beweisen, dass bipartite zyklische Segmentgraphen genau Schnittgraphen von vertikalen und horizontalen Segmenten in der Ebene sind. Hieraus können wir mit einem Ergebnis von Kratochvíl [36] folgen, dass es ein NP-vollständiges Problem ist zu testen, ob ein Graph ein zyklischer Segmentgraph ist. In späteren Abschnitten beschäftigen wir uns mit einer Unterklasse von zyklischen Segmentgraphen, die wir *Hookgraphen* nennen. Auch Cantanzaro et al. [9] haben Hookgraphen unabhängig von uns untersucht und wurden durch ein Problem in der Biologie motiviert. Ein *Hookgraph* ist ein Segmentgraph, der eine Segmentdarstellung hat, dessen Segmente alle tangential zu einer Parabel liegen und keine zwei Segmente parallel sind. In dieser Arbeit führen wir Hookgraphen mit Hilfe eines anderen Modells von zyklischen Segmentgraphen ein. Von diesen anderen Modellen beweisen wir auch, dass ein Graph ein Hookgraph ist, genau dann, wenn es der Schnittgraph von achsenparallelen Rechtecken in der Ebene ist, wobei die linke obere Ecke von jedem Rechteck auf der Diagonalen $\{(x, x) : x \in \mathbb{R}\}$ liegt. Wir führen die cross completion property für eine Anordnung von Knoten ein. Wir zeigen, dass sich Hookgraphen als Graphen charakterisieren lassen, für die eine Anordnung von Knoten existiert, welche diese cross completion property erfüllt. Mit Hilfe dieser verschiedenen Modelle, zeigen wir, dass Intervallgraphen, außerplanare Graphen und 2DORGs alle Hookgraphen sind. Im letzten Abschnitt über diese Thema geben wir Algorithmen mit polynomialer Laufzeit zur Berechnung der Cliquenzahl und der Unabhängigkeitszahl von Hookgraphen. Wir zeigen auch, dass wir für jeden Hookgraphen G die chromatische Zahl $\chi(G) = O(\log(\Omega(G)))$ haben, wobei $\chi(G)$ und $\Omega(G)$ die chromatische Zahl von G beziehungsweise die Cliquenzahl von G ist. Wir schließen Kapitel 2 mit einem Beweis ab, der zeigt, dass die Unabhängigkeitszahl eines Hookgraphen G eine 2-Approximation der Cliquenzerlegungszahl von G ist.

Das andere Thema meiner Arbeit wird in Kapitel 3 behandelt. Hier beschäftigen wir uns mit einem Färbungsproblem, welches mit konfliktfreien Färbungen von Punktmengen verbunden ist. Sei $\mathcal{R}_L$ die Menge aller achsenparallelen Rechtecke in $\mathbb{R}^2$, die eine feste horizontale Linie L schneiden. Wir bezeichnen mit $\chi_k(\mathcal{R}_L)$, die Minimalanzahl von Farben, so dass es für jede Punktmenge P eine Färbung der Punkte gibt, so dass für jedes Rechteck $R \in \mathcal{R}_L$ mit $|R \bigcap P| \geq k$ ($k \geq 2$) zwei Punkte in $P \bigcap R$ mit verschiedenen Farben gefärbt sind. Wir beweisen für $k = 2, \ldots, 5$, dass die obere Grenze von Keszegh [34] auf $\chi_k(\mathcal{R}_L)$ bestmöglich ist. Zusammen mit den Ergebnissen von Keszegh [34] ist $k = 6$ der einzige Wert von k, wofür $\chi(\mathcal{R}_L)$ unbekannt bleibt.

Bibliography

[1] D. Ajwani, K. Elbassioni, S. Govindarajan, and S. Ray. *Conflict-Free Coloring for Rectangle Ranges Using $O(n.382 + \epsilon)$ Colors.* Proceedings of the 19th annual ACM symposium on Parallel algorithms and architectures, pages $181 - 187$, (2007).

[2] E. Asplund and B. Grünbaum. *On a coloring problem.* Discrete and Computational Geometry - The Goodman-Pollack Festschrift, Math. Scand., volume 8, pages $181 - 188$, (1960). Springer.

[3] T. Asano and H. Imai. *Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane.* Journal of Algorithms, volume 4, pages $310 - 323$, (1983). Elsevier.

[4] S. Benzer. *On the topology of the genetic fine structure.* Proc. Nat. Acad. Sci. U.S.A. 45, pages $1607 - 1620$, (1959).

[5] S. Bellantoni, I. Ben-Arroyo Hartman, T. Przytycka, and S. Whitesides. *Grid intersection graphs and boxicity.* Discrete Mathematics, Volume 114, Issues $1 - 3$, Pages $41 - 49$, (1993).

[6] C. Berge. *Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung).* Wiss. Z. Natrin Luther Univ. Math.-Natur. Reihe 10, pages $114 - 115$, (1961).

[7] K.S. Booth and G.S. Lucker. *Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree Algorithms.* J. Comput. Systems Sci., 13, pages $255 - 265$, (1976). Elsevier.

[8] S. Cabello, J. Cardinal, and S. Langerman. *The clique problem in ray intersection graphs.* Algorithms - ESA - 20th Annual European Symposium, volume 7501 of Lecture Notes in Computer Science, pages $241 - 252$, (2012). Springer Verlag.

[9] D. Cantanzaro, S. Chaplick, B. Halldórsson, M. Halldórsson, and J. Stacho. *Max Point-Tolerance Graphs.* Preprint.

[10] P. Chalermsook. *Coloring and Maximum Independent Set of Rectangles.* Lecture Notes in Computer Science, volume 6845, pages $123 - 134$, (2011).

[11] P. Chalermsook and J. Chuzhoy. *Maximum Independent Set of Rectangles.* SODA'09, pages $892 - 901$, (2009).

[12] J. Chalopin, D. Gonçalves, and P. Ochem. *Planar graphs are in 1-string.* Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete Algorithms, pages $609 - 617$, (2007).

[13] J. Chalopin and D. Gonçalves. *Every planar graph is the intersection graph of segments in the plane.*. Proceedings of the 41st ACM-Symp on Theory of Computing, STOC, pages 631 – 638, (2009).

[14] X. Chen, J. Pach, M. Szegedy, and G. Tardos. *Delaunay graphs of point sets in the plane with respect to axis-parallel rectangles.* SODA '08, Proceedings of the nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 94 – 101, (2008).

[15] V. Chepoi1 and S. Felsner. *Approximating hitting sets of axis-parallel rectangles with opposite corners separated by a monotone curve.* In review.

[16] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. *The strong perfect graph theorem.* Annals of Mathematics, volume 164(1), pages 51 – 229, (2006).

[17] S.A. Cook. *The complexity of theorem-proving procedures.* Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, pages 151 – 158, (1971).

[18] D.G. Corneil, S. Olariu, and L. Stewart. *The LBFS Structure and Recognition of Interval Graphs.* SIAM Journal on Discrete Mathematics, volume 23(4), pages 1905 – 1953, (2010).

[19] R. Diestel. *Graph Theory (Graduate Texts in Mathematics), 4th edition*, (2012). Springer Verlag.

[20] G. Ehrlich, S. Even, and R.E. Tarjan. *Intersection graphs of curves in the plane.* J. Combin. Theory Series B, volume 21, pages 8 – 20, (1976). Elsevier.

[21] G. Even, Z. Lotker, D. Ron, and S. Smorodinsky. *Conflict-Free Colorings of Simple Geometric Regions with Applications to Frequency Assignment in Cellular Networks.* SIAM J. Comput. volume 33(1), pages 94 – 136, (2003).

[22] T. Feder, P. Hell, and J. Huang. *List homomorphisms and circular arc graphs.* Combinatorica, volume 19, pages 487 – 505, (1999). Springer Verlag.

[23] S. Felsner. *Geometric Graphs and Arrangements*, (2004). Springer Verlag.

[24] D.G. Fon-Der-Flaass and A.V. Kostochka. *Covering boxes by points.* Discrete Math. volume 120, pages 269 – 275. (1993).

[25] R. Fowler, M. Paterson, and S.L. Tanimoto. *Optimal Packing and Covering in the Plane are NP-Complete.* Inf. Process. Lett. volume 12, pages 133 – 137. (1981).

[26] P.C. Gilmore and A.J. Hoffman. *A characterization of comparability graphs and of interval graphs.* Can. J. Math. volume 16, pages 539 – 548, (1964).

[27] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Second edition, Annals of Discrete Mathematics, volume 57, (2004). Elsevier.

[28] S. Har-Peled and S. Smorodinsky. *On Conflict-free Coloring of Points and Simple Regions in the Plane.* Discrete and Computational Geometry (DCG), volume 34, pages 47 – 70, (2005).

[29] M. Habib, R. McConnell, C. Paul, L. Viennot. *Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing.* Theoretical Computer Science volume 234(1 − 2), pages 59 − 84, (2000).

[30] W. Haken and K. Appel. *Solution of the Four Color Map Problem.* Scientific American, volume 237(4), pages 108 − 121, (1977).

[31] E. Helly. *Über Mengen konvexer Körper mit gemeinschaftlichen Punkten.* Jahresbericht Deutsch. Math. Vereinig. volume 32, pages 175 − 176, (1923).

[32] C. Hendler. *Graphenklassen mit durch Funktionen in der Cliquezahl beschränkter Färbungszahl* . Diplomarbeit. (1998).

[33] G. Károlyi. *On point covers of parallel rectangles.* Periodica Math. Hung. volume 23, pages 105 − 107.(1991).

[34] B. Keszegh. *Combinatorial and computational problems about points in the plane.* PhD thesis, (2009).

[35] A. Kostochka. *Coloring intersection graphs of geometric figures with a given clique number.* Contemporary Mathematics: Towards a Theory of Geometric Graphs, pages 127 − 138, (2004).

[36] J. Kratochvíl. *A special planar satisfiability problem and a consequence of its NP-completeness.* Discrete Applied Mathematics, volume 52, pages 233 − 252, (1994). Elsevier.

[37] J. Kratochvíl and J. Matoušek. *Intersection graphs of segments.* Journal of Combinatorial Theory Series B, volume 62(2), pages 289 − 315, (1994). Elsevier.

[38] J. Kratochvíl and A Kuběna. *On intersection representations of co-planar graphs.* Discrete Mathematics. volume 178, pages 251 − 255, (1998).

[39] J. Kratochvíl. *String graphs. II. recognizing string graphs is NP-hard.* Journal of Combinatorial Theory, Series B, volume 52, (1991). Elsevier.

[40] K. Kuratowski. *Sur le problème des courbes gauches en topologie.* Fund. Math. 15, pages 271 − 283, (1930). Elsevier.

[41] J. L. Lovász. *A characterization of perfect graphs.* Journal of Combinatorial Theory, Series B, volume 13(2), pages 95 − 98, (1972). Elsevier.

[42] R. McConnell. *Linear-time recognition of circular-arc graphs.* Algorithmica, volume 37(2), pages 93 − 147, (2003).

[43] J.I. Moore and W.T. Trotter. *Characterization problems for graphs, partially ordered sets, lattices, and families of sets.* Discrete Mathematics, volume 16, pages 361 − 381. (1976).

[44] J. Mycielski. *Sur le coloriage des graphes.* Colloq. Math. volume 3, pages 161 − 162, (1955).

[45] J. J. Pach; P.K. Agarwal. *Combinatorial geometry.* (1995). New York: Wiley-Interscience.

[46] J. Pach, G. Tardos, and G. Tóth. *Indecomposable coverings.* Discrete Geometry, Combinatorics and Graph Theory, The China-Japan Joint Conference (CJCDGCGT 2005), Lecture Notes in Computer Science, volume 4381, pages 135 – 148, (2007). Springer Verlag.

[47] J. Pach and G. Tóth. *Conflict-free Colorings.*Discrete and Computational Geometry - The Goodman-Pollack Festschrift, pages 665 – 671, (2003), Springer Verlag.

[48] J. Pach and G. Tóth. *Recognizing string graphs is decidable.* Graph Drawing, Lecture Notes in Computer Science, pages 247 – 260, (2001). Springer.

[49] A. Pawlik, J. Kozik, T. Krawczyk, M. Lason, P. Micek, W.T. Trotter, and B. Walczak. *Triangle-free intersection graphs of line segments with large chromatic number.* arXiv:1209.1595, (2012).

[50] F.S. Roberts. *On the boxicity and cubicity of a graph.* Recent Progress in Combinatorics, Academic Press, pages 301 – 310. (1969).

[51] N. Robertson, D.P. Sanders, P. Seymour, R. Thomas *The Four-Colour Theorem.*Journal of Combinintorial, Theory Ser. B, volume 70(1), pages 2 – 44, (1997). Elsevier.

[52] M. Schaefer, E. Sedgwick, and D. Štefankovič. *Recognizing string graphs in NP.* Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, pages 19 – 21, May, (2002).

[53] E. Scheinerman. *Intersection classes and multiple intersection parameters of graphs.* PhD thesis, Princeton University, (1984).

[54] A.M.S. Shrestha, S. Tayu, and S. Ueno. *Orthogonal ray graphs and nano-pla design.* Proceedings of the IEEE International Symposium on Circuits and Systems, pages 2930 – 2933, (2009).

[55] A.M.S. Shrestha, Y. Kobayashi, S. Tayu, and S. Ueno. *On orthogonal ray graphs.* IPSJ SIG Technical Reports, volume 2008, no. 84, pages 9 – 15, (2008).

[56] A.M.S. Shrestha, S. Tayu, and S. Ueno. *On orthogonal ray graphs.* Discrete Applied Mathematics, volume 158(15), pages 1650 – 1659, (2010).

[57] S. Smorodinsky. *Conflict-Free Coloring and its Applications.* arXiv:1005.3616v3

[58] J. Spencer. *Túran's theorem for k-graphs.* Discrete Mathematics, volume 2, pages 183 – 186, (1972).

[59] W.T. Trotter. *Combinatorics and Partially Ordered Sets: Dimension Theory.* 1992. The John Hopkins University Press.

[60] G. Wegner. *Über eine kombinatorisch-geometrische Frage von Hadwiger and Debrunner.* Israel J. Math., volume 3, pages 187 – 198. (1965).

[61] M. Yannakakis. *The complexity of the partial order dimension problem.* SIAM Journal on Algebraic and Discrete Methods, volume 3(3), pages $351 - 358$, (1982).