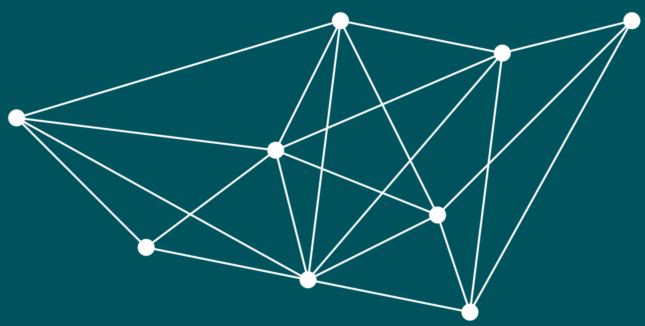
Ph.D. Thesis

ON SUPPORT AND RECOGNITION PROBLEMS FOR SPARSE HYPERGRAPHS

Karamjeet Singh



Indraprastha Institute of Information Technology, Delhi
New Delhi, India

2026

On Support and Recognition Problems for Sparse Hypergraphs

A Thesis

submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Karamjeet Singh

(PhD19307)

DEPARTMENT OF MATHEMATICS

Department of Mathematics,

Indraprastha Institute of Information Technology, Delhi

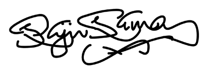
New Delhi - 110020, India

Thesis Evaluation Committee

The following experts evaluated this thesis and served as examiners.

Advisor

Rajiv Raman
Associate Professor, Dept. of Mathematics,
Dept. of Computer Science and Engineering,
IIIT Delhi



External Examiner

Hubert Chan
Associate Professor, Dept. of Computer Science,
University of Hong Kong

External Examiner

L. Sunil Chandran
Professor, Dept. of Computer Science and Automation,
Indian Institute of Science, Bangalore.

External Examiner

Arijit Ghosh
Associate Professor, Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata.

Declaration

I hereby declare that the work presented in this thesis, entitled **On Support and Recognition Problems for Sparse Hypergraphs**, is original and has been carried out by me under the supervision of **Dr. Rajiv Raman**, Associate Prof, Dept. of Mathematics and Dept. of Computer Science and Engineering, **IIIT-Delhi**. This thesis has not been submitted, either in full or in part, for the award of any other degree or diploma at this or any other institute or university. All sources of information used in this thesis have been duly acknowledged, and where the work of others has been used, it has been properly cited.

Karamjeet Singh

Karamjeet Singh
PhD student
Dept. of Mathematics,
IIIT-Delhi, India.

Place: New Delhi
Date: 13^{th} January 2026

Thesis Certificate

This is to certify that the thesis titled **On Support and Recognition Problems for Sparse Hypergraphs**, submitted by **Karamjeet Singh**, to the Indraprastha Institute of Information Technology, Delhi, for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work done by him under my supervision. The content of this thesis, in full or in parts, has not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Rajiv Raman
Thesis Supervisor
Associate Professor
Dept. of Mathematics, and
Dept. of CSE, IIIT-Delhi, India.

Place: New Delhi
Date: 13^{th} January 2026

Dedicated to

The Almighty and my family

ਆਗਾਹਾ ਕੂ ਤ੍ਰਾਘਿ ਪਿਛਾ ਫੇਰਿ ਨ ਮੁਹਡੜਾ ॥

(*Sri Guru Granth Sahib Ji - Ang 1096*)

"Having set your goals, don't turn back."

Acknowledgement

At the commencement of this acknowledgement, I would like to express my deepest gratitude to my PhD advisor, Professor Rajiv Raman, for his unwavering guidance, encouragement, and support throughout my doctoral studies. His insightful suggestions, patience, and trust in my work played a central role in shaping my research journey. I feel fortunate to have had the opportunity to learn from him and work under his supervision.

IIIT-Delhi, as my host institute, provided the academic environment, infrastructure, and facilities that made this research possible. I also acknowledge the encouragement and institutional support offered by the department and the institute throughout my PhD. The opportunities and academic culture at IIIT-Delhi have been instrumental in my development as a researcher.

A particularly enriching phase of my doctoral journey was my research visit to the *Alfréd Rényi Institute of Mathematics, Budapest*. I sincerely thank Prof. Dömötör Pálvölgyi for hosting me there during the special semester on *Discrete Geometry and Convexity* held by the *Erdős Center* in Budapest. This visit was academically rewarding, and I greatly appreciate this opportunity to meet and collaborate with prominent researchers in discrete mathematics from around the world.

With the completion of this thesis, I would like to acknowledge Prof. Hubert Chan (University of Hong Kong), Prof. Sunil Chandran (IISc, Bangalore), and Prof. Arijit Ghosh (ISI, Kolkata) for their careful and rigorous evaluation of this thesis. I am indebted to them for their valuable time and feedback. Their constructive comments strengthened the overall presentation of this thesis.

I sincerely thank Prof. Sankha S. Basu, Prof. Syamantak Das, and Prof. Samrith Ram of IIIT-Delhi for being part of my PhD internal committee, and monitoring my research progress through annual evaluations, valuable feedback and suggestions. I also thank Prof. Saurabh Ray (NYU Abu Dhabi) for serving on my fellowship upgradation committee. Additionally, being a teaching assistant of Prof. Samaresh Chatterji for several semesters, his academic discipline and principled approach helped me develop responsibilities and confidence in my academic duties. I remember him with deep respect and gratitude, as well as the happy moments we shared enjoying various music concerts in New Delhi. I deeply mourn his passing in June 2024.

Behind the scenes, the smooth functioning of academic life depends heavily on institutional support. I gratefully acknowledge the assistance of the administration, academic, finance and IT staff of IIIT-Delhi for their timely help with official processes, infrastructure, and technical resources. Equally important is the contribution of the cleaning and housekeeping staff, whose consistent efforts keep the campus clean, organized, and welcoming, thereby sustaining a productive academic environment.

Beyond the formal academic structure, friendships and collegiality often define the true experience of a PhD. My time at IIIT-Delhi was made both enjoyable and productive by the support of my labmates and colleagues. In particular, I would like to thank Sagnik and Yash for their constant support and many helpful discussions, both academically and personally. Their encouragement helped me stay motivated during challenging times. My numerous discussions with Suryendu, Tharrmashastha, and Ritesh supported me immensely and contributed significantly to my growth. A big thanks to my labmates - Arpit, Saurav, Sourav, Debasish, Purbasha, and other friends - for their companionship and contributions to a positive research environment.

No acknowledgement would be complete without recognizing the role of family, which remains the most enduring source of strength. Above all, I am deeply grateful to my parents for their unconditional love, support, and encouragement, as well as their constant belief in me, patience, and sacrifices that made it possible for me to pursue higher studies with confidence and focus. Their values, guidance, and support have been the foundation of everything I have achieved. I am grateful to my sisters, whose support played a crucial role in enabling me to pursue higher education after my schooling. Additionally, I would like to express my sincere gratitude to my brother, Bohar Singh, whose support has been invaluable throughout my studies. He consistently took responsibility for family matters and related duties, ensuring that I remained free from distractions and could devote myself fully to my research. His encouragement, care, and steady support played a major role in helping me complete this work with peace of mind and determination.

Finally, my special gratitude is due to Mr. Tar Singh and Sant Hari Singh, who have been very close to me for their constant encouragement, positive advice, and support during both the good and difficult phases of my life. Their words of motivation and steady guidance helped me remain optimistic and focused throughout my Doctorate. I remain grateful for their continued presence and encouragement, which have been a source of strength at important moments of my academic and personal life.

Karamjeet Singh

Karamjeet Singh

13^{th} January 2026

Abstract

A hypergraph $\mathcal{H}$ is a pair $(V, \mathcal{E})$, where V is a set of vertices, and $\mathcal{E}$ is a collection of subsets of V, called hyperedges. They are used to express complex relations, and they generalize graphs where each element of $\mathcal{E}$ is a 2-element subset of V. Hypergraphs are one of the most important combinatorial objects of study in theoretical computer science, and have applications in several domains, including network design, scheduling problems, biology, machine learning, etc. Thus, it is important to study their structural properties.

Starting with the work of Zykov [Zyk74], Voloshina and Feinberg [VF84], and Johnson and Pollack [JP87], researchers have made several attempts to study the structure of a hypergraph by associating with it an appropriate graph. While their initial attempts were to introduce the planarity of a hypergraph, the notion developed in [VF84; JP87] can be generalized and is now called a *support*.

A *support* for a hypergraph $\mathcal{H} = (V, \mathcal{E})$ is a graph $Q = (V, F)$ such that for each hyperedge $E \in \mathcal{E}$, the induced subgraph $Q[E]$ on the elements of E is connected. With this notion, a hypergraph is considered planar if it admits a support that is a planar graph.

The concept of support has practical applications in hypergraph visualization, network design, and several optimization problems. Although deciding whether a hypergraph admits a planar support is NP-hard, identifying sufficient conditions for the existence of such supports, particularly sparse or structured ones, remains a compelling research direction. Most of this thesis delves into the construction of supports for various graph classes.

This thesis is divided into three parts.

In Part (A), we consider hypergraphs defined by subgraphs of a given host graph. Let $G = (V, E)$ be a graph and $\mathcal{H}$ be a collection of subgraphs of G. Then the pair $(G, \mathcal{H})$ naturally defines a hypergraph with vertex set V and a hyperedge $V(H)$ for each $H \in \mathcal{H}$. We study support construction in three different settings, depending on whether the host graph G belongs to the class of graphs of (i) bounded genus, (ii) outerplanar, or (iii) bounded treewidth. We gave sufficient conditions that ensure the existence of a support from the same family of graphs as G. The results are extended to *dual* hypergraphs and to a more general setting- the *intersection hypergraphs*. We also present a fast algorithm for the construction of a planar support with straight-line embedding when the underlying hypergraph is defined by axis-parallel rectangles and points in $\mathbb{R}^2$.

Part (B) of the thesis explores the role of supports in solving classical problems such as packing, covering, and coloring problems in hypergraphs. We study these problems for hypergraphs arising from subgraphs of a host graph as well as from geometric regions on orientable surfaces, and present approximation results to the packing and covering problems above.

Finally, Part (C) turns to abstract hypergraphs and examines the computational complexity of identifying vertex orderings that forbid fixed patterns. We show NP-hardness of this problem for several vertex orderings, and we deduce implications for the recognition of hypergraphs defined by geometric regions in $\mathbb{R}^2$.

Publications

The majority of the results in this thesis are based on the following papers.

[Dam+25] Gábor Damásdi, Balázs Keszegh, Dömötör Pálvölgyi, and Karamjeet Singh. *The complexity of recognizing ABAB-free hypergraphs*. 2025. URL: https://dmtcs.episciences.org/15562.

[Pal+24] Ambar Pal, Rajiv Raman, Saurabh Ray, and Karamjeet Singh. "A Fast Algorithm for Computing a Planar Support for Non-Piercing Rectangles". In: *35th International Symposium on Algorithms and Computation (ISAAC 2024)*. Ed. by Julián Mestre and Anthony Wirth. Vol. 322. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 53:1–53:18. ISBN: 978-3-95977-354-6. DOI: 10.4230/LIPIcs.ISAAC.2024.53. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ISAAC.2024.53.

[RS23] Rajiv Raman and Karamjeet Singh. "On Hypergraph Supports". In: *Proceedings of the 12th European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB'23)*. 2023. DOI: https://doi.org/10.5817/CZ.MUNI.EUROCOMB23-107.

[RS25a] Rajiv Raman and Karamjeet Singh. *On Supports for graphs of bounded genus*. 2025. arXiv: 2503.21287 [math.CO]. URL: https://arxiv.org/abs/2503.21287.

[RS25b] Rajiv Raman and Karamjeet Singh. *Supports for Outerplanar and Bounded Treewidth Graphs*. 2025. arXiv: 2504.05039 [math.CO]. URL: https://arxiv.org/abs/2504.05039.

Table of Contents

Prologue

CHAPTER 1

Introduction

Given a set V, a *graph* on V is defined by a pair (V, E), where E is some collection of 2-element subsets of V. The elements of V are *vertices*, and those of E are called *edges* of G. Graphs are among the most fundamental combinatorial structures studied in discrete mathematics and theoretical computer science. Their versatility in modeling pairwise relationships has made them indispensable across a wide range of fields, from classical algorithm design and complexity theory to modern areas such as network science, computational biology, social network analysis, and logistics. The study of graph-theoretic properties and their structural extensions continues to play a central role in both the development of theoretical frameworks and the design of efficient algorithms for real-world problems.

Among the many structural properties of graphs, one of the most extensively studied is their embeddability, most notably, in the plane. An *embedding* of a graph $G = (V, E)$ in the plane is a drawing, where each $v \in V$ corresponds uniquely to a point $p_v \in \mathbb{R}^2$, and each edge $\{u, v\} \in E$ corresponds uniquely to a simple Jordan arc in the plane with end points p_u and p_v, which does not pass through any other point p_w for any $w \in V \setminus \{u, v\}$. A graph G is said to be *planar* if there is a drawing of G in the plane such that for each pair of edges in G, the corresponding curves defining the edges are internally disjoint. Not all graphs are planar, e.g., K_5, i.e., the graph on 5 vertices with all the $\binom{5}{2}$ possible edges. Graph planarity is well studied in the literature, originating from the work of Leonhard Euler in the 18th century.

A graph that is not planar can be embedded on other orientable surfaces of higher *genus* (formally defined in Definition 1.14). For example, K_5 is non-planar, but can be embedded on a *torus*. A graph H is a *minor* of a graph G if H can be obtained from a subgraph of G by a sequence of edge contractions. By the Graph Minor Theorem

[RS04], for any constant g, the class of graphs embeddable on a surface of genus g can be characterized by a *finite* list of excluded minors. Besides their structural properties, such graphs are *sparse* in the sense that they have $O(gn)$ number of edges, where n is the number of vertices in the graph.

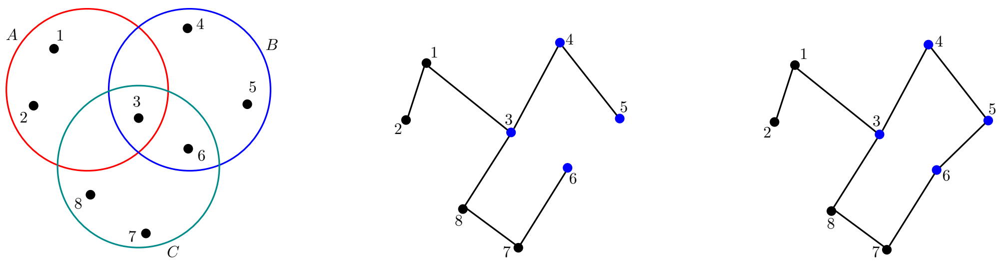
Extending the concept of graphs, hypergraphs are combinatorial structures that generalize graphs. A *hypergraph* $\mathcal{H}$ is defined by a pair $(V, \mathcal{E})$, where V is a set and $\mathcal{E}$ is a collection of subsets of V. The elements of V are called *vertices*, and those of $\mathcal{E}$ are called *hyperedges* of $\mathcal{H}$. Here, unlike for graphs, an element of $\mathcal{E}$ need not be a 2-element set.

In several problems involving graphs, the underlying graph is planar, or can be embedded on a surface of finite genus, or more generally, an H-minor free graph for some fixed graph H. This sparsity of the underlying graph can be exploited to obtain faster or better algorithms than those developed for general graphs. Similarly, in several applications, the natural objects in consideration are hypergraphs. Thus, it is desirable to establish a similar notion of *planarity* or *sparsity* for these generalized objects, particularly to get better algorithmic applications.

A natural approach to explore their structural properties is to associate with each hypergraph an auxiliary graph satisfying some specific rules, depending upon the intended notion of planarity or sparsity. In this case, we call a hypergraph planar (sparse) if the associated graph is planar (sparse). This provides a natural framework to extend these notions from graphs to hypergraphs. Several researchers have thus attempted to define a notion of planarity for hypergraphs. Unlike the case for graphs, however, there is no unique notion of hypergraph planarity. There are several incomparable notions, each arising out of the specific requirements of the corresponding application. We will discuss these in brief in the upcoming sections of this chapter. As a warm-up, consider the following as a candidate example introduced by Voloshina and Feinberg in the study of *Very Large Scale Integrated circuits* (VLSI circuits) [VF84].

A VLSI circuit may be considered as a hypergraph in which an electric component corresponds uniquely to a vertex of the hypergraph, and a particular set of components, called a *net*, corresponds uniquely to a hyperedge. The problem is to connect the components of the circuit with wires so that within every net, we can reach from one component to another via a sequence of wires and components such that no two wires of the circuit cross each other. In other words, we require a planar graph Q whose vertex set corresponds to the set of components of the circuit, and edges correspond to the wires added to the circuit such that for each net, its components induce a *connected subgraph* in Q. The authors call a hypergraph planar if there exists such a planar graph Q satisfying the connectivity requirement within each net (hyperedge).

The hypergraph planarity notion introduced by Voloshina and Feinberg is widely accepted as it generalizes other similar notions introduced by different authors, and is called a support in the present context. Formally, A *support* for a hypergraph $\mathcal{H} = (V, \mathcal{E})$ is a graph Q on V such that $Q[E]$, the subgraph of Q induced on the vertices in E, is connected for all $E \in \mathcal{E}$. See Fig. 1.1 for a basic example. In Fig. 1.1a A hypergraph is defined on vertices $1, 2, \ldots, 8$. Each of the disks A, B and C defines a hyperedge consisting of the vertices contained in it. Fig. 1.1b shows a graph on these vertices, which is not a support since the vertices of the hyperedge B (blue vertices) do not induce a connected subgraph. Finally, Fig. 1.1c shows a graph that is a support since it satisfies the connectivity requirement for each hyperedge.



(a) Hypergraph **(b)** Not a support **(c)** A support graph

Figure 1.1: A hypergraph and its support.

This thesis delves into this foundational notion of support for hypergraphs, offering new insights into this fundamental yet underexplored concept and its role in the structural theory of hypergraphs.

Throughout this thesis, we consider hypergraphs that are defined on a finite set of vertices.

Table of Contents

§ 1.1 Background

Hypergraphs are more general objects than graphs, and hence, results about their structure are harder to characterize than those of graphs. One approach is by associating a graph with the hypergraph that captures some of its structure. Below, we start with a notion of planarity of hypergraphs. In contrast to graphs, the notion of planarity in hypergraphs remains largely unexplored and lacks a universally accepted definition in the literature.

The notion of a planar analogue in hypergraphs was first suggested by Zykov [Zyk74], who defined a hypergraph to be planar if there is a plane graph on the elements of the hypergraph such that for each hyperedge, there is a bounded face of the embedding containing only the elements of this hyperedge. Equivalently, a hypergraph is *Zykov-planar* iff its incidence bipartite graph is planar.

Voloshina and Feinberg [VF84] introduced another notion of hypergraph planarity in the context of planarizing VLSI circuits (see the monograph by Feinberg et al. [FLR12] and references therein). Johnson and Pollak [JP87] introduced two notions of hypergraph planarity, namely *vertex-based Venn diagram* and *edge-based Venn diagram*. They call a hypergraph *vertex-planar* if it admits a vertex-based Venn diagram. This notion is equivalent to the planarity notion introduced by Voloshina and Feinberg, and is now called a *planar support*, i.e., a support that is a planar graph. This generalizes the planarity in graphs, i.e., if G is a planar graph, then as a hypergraph, G admits a planar support. The edge-minimal non-planar graph is the complete bipartite graph $K_{3,3}$, which contains exactly nine edges. Building on this result, Verroust and Viaud [VV04] extended the notion to hypergraphs, demonstrating that every hypergraph with at most eight hyperedges admits a planar support. Generalizing the concept of planarity, we can say that a hypergraph comes from a family $\mathcal{F}$ if it admits a support $Q \in \mathcal{F}$.

There are other notions of hypergraph drawings in the plane studied in the literature, namely, *subset-based, edge-based, concrete Euler diagrams*, and *simple* and *compact subdivision drawings*. We refer the reader to [KKS09] and references therein for more details.

§ 1.1.1 Sparse support graphs

Any hypergraph $\mathcal{H} = (V, \mathcal{E})$ clearly has a support - e.g., a complete graph on V, or a union of connected graphs, each corresponding uniquely to the vertices of a hyperedge E, for all $E \in \mathcal{E}$. The problem becomes interesting if we introduce *global* constraints on the graph that is in *tension* with the *local* connectivity requirement for each hyperedge. In particular, we are interested in a support that comes from a *sparse* family of graphs. In general, it is not always possible to have such a support, e.g., consider the hypergraph defined on an n-element set V, where the hyperedges are defined by all the 2-element subsets of V. Clearly, the only support is K_n, a complete graph on n vertices. So, which hypergraphs admit a support that is a sparse graph, such as a planar graph? Can we decide *efficiently* if a hypergraph exhibits a sparse support? To proceed, we restate the problem in a more general and mathematically rigorous formulation, as captured in the following question.

> **Question 1.1.** Given a hypergraph $\mathcal{H}$ and a family of graphs $\mathcal{G}$, is it possible to decide *efficiently* if $\mathcal{H}$ admits a support Q such that $Q \in \mathcal{G}$?

First, we explain a bit about what "efficiently" means in this particular context. Accordingly, consider the input hypergraph $\mathcal{H} = (V, \mathcal{E})$. If $|V| = n$ and $|\mathcal{E}| = m$, then we define the *size* of $\mathcal{H}$ to be $\mathcal{S}_{\mathcal{H}} = N + n + m$, where $N = \sum_{H \in \mathcal{H}} |H|$. We say that we can solve Question 1.1 efficiently if there is an algorithm to solve the corresponding decision problem in time polynomial in $\mathcal{S}_{\mathcal{H}}$.

Johnson and Pollack [JP87] answered this question negatively, even if $\mathcal{G}$ is the class of all planar graphs. In other words, the authors proved that it is NP-complete to decide if a given hypergraph admits a support that is a planar graph. Du showed that it is NP-hard to find a support with the minimum number of edges [Du86]. Since then, several authors have studied Question 1.1 for various graph classes $\mathcal{G}$.

Tarjan and Yannakakis [TY84] showed that we can decide in linear time if a hypergraph admits a tree support. Buchin et al., [Buc+11] showed linear time algorithms to decide if a hypergraph admits a support that is a path or a cycle, and a polynomial time algorithm to decide if a hypergraph admits a support that is a tree with bounded maximum degree. Further, the authors sharpen the result of Johnson and Pollak [JP87] by showing that deciding if a hypergraph admits a support that is a k-

outerplanar graph[1], is NP-hard for each $k \geq 2$. The notion of constructing a support with minimum maximum degree has also been studied in [Bal+07a; OR11].

A graph is a *cactus* if each of its edges participates in at most one cycle of the graph. Brandes et al. [Bra+11] showed that it can be tested in $O(nN + n + m)$ time if a hypergraph admits a cactus support. With the same running time, one can test if there is a planar or outerplanar support for a hypergraph closed under intersections and differences (hcid). A *path-based support* for a hypergraph is a support such that for each hyperedge, there is a *Hamiltonian path* on the vertices of that hyperedge. It is NP-hard to decide if a hypergraph has a planar path-based support; however, a path-based tree support can be constructed in polynomial time provided it exists [Bra+12]. We summarise these results in Table 1.1 below.

Support graph	Complexity	References
Path, Cycle	$\mathcal{O}(N)$	[Buc+11]
Tree	$\mathcal{O}(N)$	[TY84]
Bounded-degree tree	$\mathcal{O}(n^3 + mn^2)$	[Buc+11]
Cactus	$\mathcal{O}(nN + n + m)$	[Bra+11]
Outerplanar (hcid)	$\mathcal{O}(nN + n + m)$	[Bra+11]
Path-based tree	$O(mn^3)$	[Bra+12]
Planar path-based	NP-complete	[Bra+12]
2-outerplanar	NP-complete	[Buc+11]
Outerplanar	Open	–
Compact subdivision	NP-complete	[KKS09]
Planar	NP-complete	[JP87]
Minimum no. of edges	NP-hard	[Du86]

Table 1.1: The table above shows the time complexity of deciding the existence of support in various graph classes.

The concept of support has found wide applicability in several areas, such as visualizing hypergraphs [Ber+15; Ber+11; Bra+11; Bra+12; Buc+11; Hav+22; Hur+18], in the design of networks [Anc+06; Bal+07a; Bal+07b; CF05; Hos+12; KS03; OR11], and similar notions have been used in the analysis of *local search algorithms* for geometric problems [AMM17; BR+18; CAM15; Kro+14; MR10; RR20]. For many of these applications, the requirement is the existence or construction of a sparse graph that is a support or is of a similar notion.

[1] See Definition 1.15 for the formal notion of an outerplanar graph. A graph is k-outerplanar if it has a planar embedding such that removing the vertices of the outer face results in a $(k-1)$-outerplanar graph.

Two classes of sparse graphs that have been studied intensively are those that are *easily decomposable*, i.e., graphs with *sublinear-sized balanced separators*[2], and graphs that satisfy various notions of *expansion* [HLW06]. Examples of the former are planar graphs [LT79], graphs of bounded *genus* (Definition 1.14) [GHT84], graphs excluding a minor [AST90], and more generally, graphs with polynomially bounded *t-shallow minors* [NM12]. The fact that a family of graphs have sublinear separators has been exploited to develop faster algorithms, or algorithms with better approximation factors than in general graphs. Some results that use this paradigm can be found in [LT80; Fed87; Aro97; FP14; AHPW19]. Similarly, there are examples of algorithms that exploit expansion for faster algorithms, or to obtain algorithms with better approximation factors [Alo+23; Aro+08]. In a similar vein, one would like to develop a notion of sparsity of hypergraphs that can be exploited algorithmically. The existence of a sparse support for a hypergraph is one such notion.

§ 1.2 Motivation

This section outlines the motivation behind our study of supports. In addition to their applicability in various domains, as discussed in Section 1.1.1, our primary motivation stems from several fundamental problems in hypergraph theory – specifically, those related to *packing*, *covering*, and *coloring* problems, which we elaborate in this section.

In this thesis, we consider hypergraphs defined by subgraphs of a host graph - the vertices of the hypergraph are the subsets of vertices of the host graph, and hyperedges are defined by the vertex subsets of the subgraphs at hand. In particular, we consider hypergraph that are arise from *non-piercing subgraphs* of a host graph G – a collection $\mathcal{H}$ of connected subgraphs of G is non-piercing if for each $H, H' \in \mathcal{H}$, the vertices $V(H) \setminus V(H')$ induce a connected subgraph of G. In Section 1.3, we explain this framework in more detail. The motivation behind this setup came from the so-called *non-piercing regions*, which we define below.

In the following, we assume that the geometric regions defined in the plane are in *general position*, i.e., the boundaries of no three regions pass through a common point, the boundaries of any two regions intersect a finite number of times, and their boundaries cross at the point of intersection. We need the following definition before we proceed further.

[2] A graph has a sublinear-sized balanced separator if there are constants $\epsilon > 0$, and $c > 0$ and a set S of size $O(|V|^{1-\epsilon})$ such that $G \setminus S$ contains two disconnected components A and B such that $|A|, |B| \leq c|V|$.

Definition 1.1 (Non-piercing regions)**.** Two connected regions $R, R' \subseteq \mathbb{R}^2$ are said to be *non-piercing* if both $R \setminus R'$ and $R' \setminus R$ are connected subsets of $\mathbb{R}^2$. A family $\mathcal{R}$ of connected regions in $\mathbb{R}^2$ is a family of non-piercing regions (or a non-piercing family) if the regions in $\mathcal{R}$ are pairwise non-piercing.

If, in addition, the regions in $\mathcal{R}$ are assumed to be *simply connected*[3], and boundaries of any two of them intersect at most k-times, then $\mathcal{R}$ is said to be a family of *k-admissible regions* [WZ90]. Let us start with the packing and covering problems on hypergraphs.

§ 1.2.1 PTAS for packing and covering problems

The term PTAS is an abbreviation for *Polynomial Time Approximation Scheme* (see Definition 6.12), which refers to approximation algorithms for optimization problems. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$, the *Point Packing* problem is to find a maximum cardinality subset of vertices, no two of which lie in a common hyperedge. The *Set Packing* problem is defined analogously, where the role of vertices and hyperedges is interchanged. In the *Set Cover* problem, the goal is to find a subset of hyperedges with the minimum cardinality so that every vertex lies in at least one of the chosen hyperedges. The *Hitting Set* problem is defined similarly by interchanging the roles of vertices and hyperedges. See Chapter 6 for formal definitions of these problems.

For a concrete example, consider the Set Packing problem for a hypergraph defined by a set $\mathcal{D}$ of disks in the plane, where the vertex set of the hypergraph is the entire Euclidean plane. Two disks $D_1, D_2 \in \mathcal{D}$ are independent if $(D_1 \cap D_2) \cap \mathbb{R}^2 = \emptyset$. The goal is to find a subset $\mathcal{D}' \subseteq \mathcal{D}$ of maximum cardinality such that disks in $\mathcal{D}'$ are pairwise independent. Hochbaum and Mass [HM85] gave a *shifting strategy* that leads to a PTAS for this problem. The technique works as long as the geometric regions are fat - a region $R \subseteq \mathbb{R}^2$ is *fat* if its lengths in two dimensions are comparable, i.e., the ratio of the side of a smallest square enclosing R with that of a largest square enclosed inside R is constant. However, the shifting strategy fails if the vertex set $\mathbb{R}^2$ is replaced by an arbitrary point set $P \subsetneq \mathbb{R}^2$.

Chan and Har-Peled [CH12] considered the Independent Set (Set Packing) problem, and gave PTAS for the hypergraphs defined by pseudodisks (2-admissible regions) in $\mathbb{R}^2$. Independently, Mustafa and Ray [MR10] gave a PTAS for the Hitting Set problem for a set of points and pseudodisks in the plane (both results apply more generally to k-admissible regions). Gibson and Pirwani [GP10] gave a PTAS for the

[3] A connected region $R \subseteq \mathbb{R}^2$ is simply connected if every Jordan curve inside R can be continuously contracted to a point while remaining inside R throughout the contract

Dominating Set problem[4] on the intersection graph of disks in the plane. The analysis in all these results used the *local-search* framework, which depends on the existence of a *local search graph* that satisfies the *local property* and has *sublinear-sized balanced separators* [GHT84; AST90]. See Chapter 6 for more details on the local search framework.

Cabello and Gajser [CG14a], and Ashner et al. [Asc+13] studied the packing and covering problems on non-planar graphs that exclude a fixed minor. The local search yielded PTAS, and the authors used the fact that the underlying H-minor-free graphs have sublinear separators [AST90].

The results in [CH12; MR10; GP10] were extended by Basu-Roy et al., [BR+18] who used the same framework to design a PTAS for the Set Cover and Dominating Set problems for the intersection graph of pseudodisks. Raman and Ray [RR20] generalized the previous approaches and showed that the *intersection hypergraph* (formally defined in Definitions 1.11 and 6.19) of non-piercing regions (not necessarily simply connected) in the plane admits a planar support. Their result gave a unified analysis for the packing and covering problems described above. The fact that the existence of a planar support implies these better approximation results for the problems above is that the support constructed satisfies both the condition required for a local search graph *viz.* the local search condition, and being a planar graph, it admits a sublinear-sized balanced separator.

Thus far, most of the research has been restricted to geometric hypergraphs in the plane (some of the results extend to halfspaces in $\mathbb{R}^3$), but nothing is known for the regions defined on non-planar surfaces. Motivated by these optimization problems, we want to go beyond planarity. One natural approach is to extend the unified framework of Raman and Ray to geometric hypergraphs defined on surfaces to get sparse supports, and then use the fact that they admit *small* separators. Unfortunately, the non-piercing condition fails for the existence of a sparse support even if the regions are defined on a torus (see Example 2.1).

This motivates us to go beyond geometry, and thus, we consider hypergraphs defined by connected subgraphs of a given host graph. We define this graph-theoretic framework in Section 1.3. A broader problem of interest is described in the following question.

> **Question 1.2** (Informal). Given a graph G and a collection $\mathcal{H}$ of subgraphs of G. What property P should the subgraphs satisfy so that the underlying hypergraph admits a sparse support?

[4] In the Dominating Set problem on a graph $G = (V, E)$, the goal is to find a subset $V' \subseteq V$ of minimum size such that each vertex in $V \setminus V'$ has at least one neighbor in V'.

It turns out that if we model these hypergraphs by subgraphs of a host graph that satisfy a *cross-free* condition (which generalizes non-piercing condition in the plane, see Definition 2.4), then we can show the existence of a support of bounded genus. This result, combined with the fact that graphs of bounded genus have sublinear-sized separators [AST90; GHT84], implies PTAS for the above packing and covering problems defined by *appropriate* regions on oriented surfaces of bounded genus. Thus, our framework generalizes the previous results in the plane as well as on surfaces of bounded genus.

We also consider two other settings - when the host graph is outerplanar, or has bounded *treewidth*. The results above can be extended to these graph classes as well.

§ 1.2.2 Sparse support and VC-dimension bounds

VC-dimension is a well-known complexity measure of a hypergraph, introduced by Vapnik and Chervonenkis (see Chapter 10 in [Mat02]). It is defined as the largest size of a subset of vertices that can be *shattered* (see Section 3.4 for a formal description). A bounded VC-dimension leads to efficient algorithms for Set Cover, Hitting Set, and other optimization problems in hypergraphs due to known approximation schemes for such hypergraphs. This also implies the existence of small ϵ*-nets* and ϵ*-approximations*. Below, we show how the existence of a sparse support results in a bounded VC-dimension. To prove this, we first introduce a *hereditary* class of hypergraphs, a notion generalizing hereditary graph classes, i.e., a graph class closed under taking induced subgraphs, which are well studied in the literature.

> **Definition 1.2** (Hereditary hypergraph class [Pá24]). Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$, a hypergraph $\mathcal{H}' = (V', \mathcal{E}')$ is a *sub-hypergraph* of $\mathcal{H}$ if $V' \subseteq V$ and $\mathcal{E}' \subseteq \mathcal{E}$. For a set $X \subseteq V$, the *restriction* of $\mathcal{H}$ to X is a hypergraph $\mathcal{H}_X = (X, \mathcal{E}_X)$, where $\mathcal{E}_X = \{X \cap E : E \in \mathcal{E}\}$. Finally, a family $\mathcal{C}$ of hypergraphs is *hereditary* if it is closed under taking sub-hypergraphs and restrictions.

Let *VC-dim*$(\mathcal{H})$ denote the VC-dimension of a hypergraph $\mathcal{H}$. For a class $\mathcal{C}$ of hypergraphs, we define *VC-dim*$(\mathcal{C}) = \max\{VC\text{-}dim(\mathcal{H}) : \mathcal{H} \in \mathcal{C}\}$. We say that $\mathcal{C}$ has *bounded* VC-dimension if for all $\mathcal{H} \in \mathcal{C}$, the *VC-dim*$(\mathcal{H}) \leq t$ for some integer t. For a graph class $\mathcal{G}$, we say that $\mathcal{C}$ possesses a $\mathcal{G}$-support if each $\mathcal{H} \in \mathcal{C}$ admits a support $Q_{\mathcal{H}} \in \mathcal{G}$. Let $\mathcal{G}(t)$ denote the class of K_t-subgraph-free graphs, i.e., for any $G \in \mathcal{G}(t)$, G does not contain K_t as a subgraph. The following result gives an upper bound on the VC-dimension of a hereditary class of hypergraphs possessing sparse supports.

Theorem 1.1. Let $\mathcal{C}$ be a hereditary class of hypergraphs possessing a $\mathcal{G}(t)$-support. Then, *VC-dim*$(\mathcal{C}) \leq t$.

Proof. If possible, let there be a hypergraph $\mathcal{H} = (V, \mathcal{E})$ in $\mathcal{C}$ such that *VC-dim*$(\mathcal{H}) \geq t$. Then, there is a set $Y = \{v_1, v_2, \ldots, v_t\} \subseteq V$ that can be shattered by $\mathcal{H}$. Therefore, for each $X \subseteq Y$, there is a hyperedge $E_X \in \mathcal{E}$ such that $E_X \cap Y = X$. Let $\mathcal{E}_Y = \{E_X \cap Y : X \subseteq Y\}$, and $\mathcal{H}_Y = (Y, \mathcal{E}_Y)$ be the restriction of $\mathcal{H}$ to Y. Since $\mathcal{C}$ is hereditary, $\mathcal{H}_Y \in \mathcal{C}$. Since each two-element subset of Y is a hyperedge in $\mathcal{H}_Y$, it follows that any support Q of $\mathcal{H}_Y$ contains K_t as a subgraph; a contradiction since $\mathcal{C}$ possesses a $\mathcal{G}(t)$ support. ■

The graph class $\mathcal{G}(t)$ captures the class of K_t-minor-free graphs. Therefore, Theorem 1.1 also holds for the K_t-minor-free class of graphs. As a consequence of Theorem 1.1, it follows that all the algorithmic results that hold for a hereditary class of hypergraphs with bounded VC-dimension can be carried out to that possessing a sparse support. A natural example of such a hereditary class is the class of geometric hypergraphs arising from points and pseudodisks in $\mathbb{R}^2$.

Note 1.2.1. The requirement "hereditary" is necessary in Theorem 1.1. In other words, if a hypergraph $\mathcal{H}$ admits a support $Q \in \mathcal{G}(t)$, it does not imply that *VC-dim*$(\mathcal{H}) < t$. For example, let $\mathcal{H} = (V, \mathcal{E})$ be the *complete* hypergraph on n vertices, and $\mathcal{H}'$ be obtained from $\mathcal{H}$ by adding a new vertex c to all the hyperedges of $\mathcal{H}$. Then a star with centre c and spokes as elements of V is a support for $\mathcal{H}'$, which is K_t-subgraph-free for all $t \geq 3$. However, *VC-dim*$(\mathcal{H}') = n$.

We will see in Chapter 2 that the converse of the argument in Note 1.2.1 is also not true, i.e., a hypergraph with bounded VC-dimension need not admit a sparse support.

§ 1.2.3 Hypergraph coloring

Another key motivation for studying sparse supports arises in the context of hypergraph coloring. A *proper* coloring of a hypergraph is a coloring of its vertices so that each hyperedge of size at least two has two vertices of different colors. It turns out that the chromatic number of a hypergraph is upper bounded by the chromatic number of any support for it, since a support has at least one edge for each hypergraph of size at least two. For example, the hypergraph defined by non-piercing regions in $\mathbb{R}^2$ is 4-colorable since it admits a planar support [RR20]. We will revisit hypergraph coloring in Section 1.5.2.2.

§ 1.3 A Novel Framework: Hypergraphs from Graph Substructures

In this section, we will explain the setup for hypergraphs that arise from subgraphs of a graph. In Chapters 2 to 4, we consider hypergraphs defined by connected subgraphs of a given host graph. This section presents a relation between the geometric and graph-theoretic setup of hypergraphs and delves into the reason and motivation behind this new conceptual framework.

As mentioned earlier, we are interested in the existence or construction of sparse supports for hypergraphs since they have applications in several domains. However, as discussed in Section 1.1.1, it is already NP-hard to decide if a hypergraph admits a support that is a planar graph [JP87] or a 2-outerplanar graph [Buc+11]. Therefore, it is natural to impose some restrictions on the hypergraphs that ensure the existence of a support from a sparse family of graphs. To that end, we start with hypergraphs defined by simple geometric structures in the plane and transform them into a combinatorial framework designed on graphs that can be potentially extended to hypergraphs defined on non-planar surfaces.

§ 1.3.1 Geometric hypergraphs

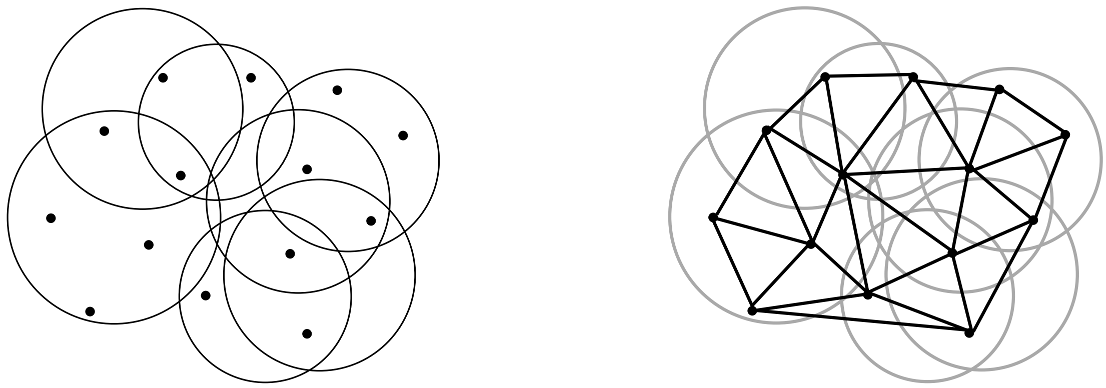
Consider a set $\mathcal{D}$ of disks and a set P of points in general position in the plane. This defines a hypergraph $(P, \mathcal{D})$ with vertex set P and hyperedges $\mathcal{D}$, where each $D \in \mathcal{D}$ defines a hyperedge consisting of vertices in $P \cap D$. Now we reformulate Question 1.1 in terms of the hypergraph at hand.

> **Question 1.3.** Given a hypergraph $\mathcal{H} = (P, \mathcal{D})$ defined by points and disks in $\mathbb{R}^2$, and a family $\mathcal{G}$ of all *planar* graphs. Can we decide *efficiently* if $\mathcal{H}$ exhibits a support $Q \in \mathcal{G}$?

The answer to the question above is *affirmative* (assuming no four points in P lie on a circle). In fact, it can be shown that for any fixed set of points P in $\mathbb{R}^2$, the *Delaunay graph*[5] on P is always a planar support for any choice of disks defining the hypergraph [KP18] (we also refer the reader to Chapter 27 of [GOT17]). See Fig. 1.2.

The result can be extended to the hypergraphs defined by points and pseudodisks in the plane [KP18]. Extending this line of work, Raman and Ray provided a polynomial-time algorithm to construct a planar support for the intersection hypergraph defined

[5] For a point set $P \subsetneq \mathbb{R}^2$, the Delaunay graph (aka Delaunay triangulation) is a graph G on P, where two vertices $p, q \in P$ are adjacent in G iff there is a disk $D_{p,q}$ through p and q that does not contain any other point P in its closure.



(a) Hypergraph of disks and points

(b) Delaunay triangulation

Figure 1.2: Delaunay triangulation is a planar support.

by non-piercing regions in the plane [RR20]. With this groundwork in place, we are ready to introduce our graph-theoretic formulation of hypergraphs.

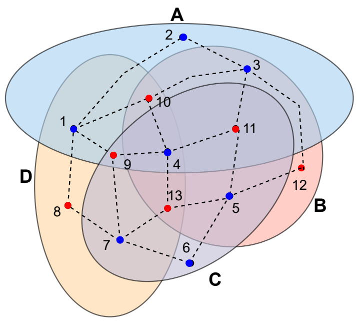
§ 1.3.2 A graph-theoretic reformulation of geometric hypergraphs

Let $\mathcal{R}$ be a collection of connected regions in the plane and $\partial(R)$ denote the boundary of a region $R \in \mathcal{R}$. Then $\mathbb{R}^2 \setminus \bigcup_{R\in\mathcal{R}} \partial(R)$ splits the plane into disjoint pieces called *cells*. A *dual-arrangement graph* of $\mathcal{R}$ is a graph that has a vertex for each cell in the arrangement of the region, and two vertices are adjacent if and only if the corresponding cells share an arc of positive length on their boundaries. Formally,

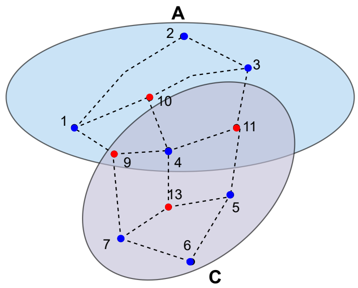
Definition 1.3 (Dual-arrangement graph)**.** Let $\mathcal{R}$ be a family of connected regions in $\mathbb{R}^2$, and $\mathcal{C}$ be the collection of cells in the arrangement of $\mathcal{R}$. Consider a set $V = \{v : C_v \in \mathcal{C}\}$. Then, the *dual-arrangement graph* of $\mathcal{R}$ is a graph $G = (V, E)$ such that for any two vertices $u, v \in V$, there is an edge $\{u, v\} \in E$ if and only the boundaries of the cells C_u and C_v share an arc of positive length.

Since the regions in $\mathcal{R}$ are connected subsets of $\mathbb{R}^2$, the dual-arrangement graph G is planar, and each region $R \in \mathcal{R}$ corresponds to a connected subgraph of G - the subgraph induced on the vertices corresponding to the cells in R. Moreover, if $\mathcal{R}$ is non-piercing, it is easy to check that for any two subgraphs R_G, R'_G corresponding to $R, R' \in \mathcal{R}$, the induced subgraphs of G on the vertices in $R_G \setminus R'_G$ and $R'_G \setminus R_G$, are connected (see Fig. 1.3b), hence the following definition.

Definition 1.4 (Non-piercing subgraphs)**.** Let G be a graph. Two subgraphs H, H' of G are said to be *non-piercing* if the sets $V(H) \setminus V(H')$ and $V(H') \setminus V(H)$ induce connected subgraphs of G. A family $\mathcal{H}$ of connected subgraphs of G is non-piercing if the subgraphs in $\mathcal{H}$ are pairwise non-piercing.



(a) Dual-arrangement graph of non-piercing regions in $\mathbb{R}^2$.



(b) Non-piercing regions correspond to non-piercing subgraphs of dual-arrangement graph.

Figure 1.3: Dual-arrangement graph on red and blue vertices and its non-piercing subgraphs defined by regions in $\mathbb{R}^2$.

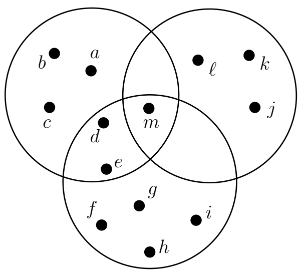
Thus, the problem of the existence or construction of a planar support for a hypergraph defined by a point set P and non-piercing regions $\mathcal{R}$ in the plane can be formulated as the corresponding graph problem in the following way.

Consider the dual-arrangement graph G of a non-piercing family of regions $\mathcal{R}$ in $\mathbb{R}^2$. We can assume without loss of generality that each cell in the arrangement of $\mathcal{R}$ that contains a point of P, contains exactly one point of P (if we can construct a planar support for this instance, we can extend it to a planar support for $(P, \mathcal{R})$ as shown in Fig. 1.4). For a cell containing a point of P, we color its corresponding vertex in G *blue*, otherwise, we color it *red*. Now, restructuring Question 1.3, we have the following problem for a planar graph:

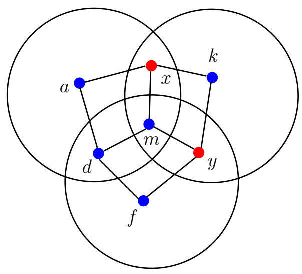
Question 1.4. Given a planar graph $G = (V, E)$, a collection $\mathcal{H}$ of non-piercing subgraphs of G, and a 2-coloring (not necessarily proper) $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. Is there a planar graph Q on the vertices $\mathbf{b}(V) = \{v \in V(G) : c(v) = \mathbf{b}\}$ such that for each $H \in \mathcal{H}$, the vertices $V(H) \cap \mathbf{b}(V)$ induce a connected subgraph of Q?

The graph G and subgraphs $\mathcal{H}$ naturally define a hypergraph on the vertices $\mathbf{b}(V)$ that has a hyperedge $V(H) \cap \mathbf{b}(V)$ for each $H \in \mathcal{H}$. We call this hypergraph a *primal hypergraph* to distinguish it from the other two hypergraphs we consider, called *dual* and *intersection* hypergraphs. For simplicity, we also call the graph Q in the question above a *primal* support for the hypergraph defined by $(G, \mathcal{H})$. We formally define all these notions in Section 1.4. The question above is answered positively in a much more general setting, where the subgraphs satisfy the condition of being cross-free (Definition 2.4), and this potentially helps in obtaining a support of bounded genus if the genus of the host graph G is bounded. In this thesis, we give sufficient conditions on the hyperedges of a hypergraph, i.e., on the subgraphs of G, that ensure the existence of an appropriate sparse support graph. Mathematically, we will be studying the following philosophical problem.

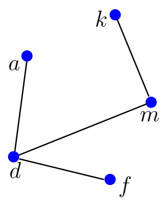
Question 1.5. Given a graph $G \in \mathcal{G}$ for some graph class $\mathcal{G}$, and a 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. Let $\mathcal{H}$ be a collection of connected subgraphs of G satisfying some property $\mathcal{P}$. Does $\mathcal{H}$ admit a support Q such that $Q \in \mathcal{G}$?



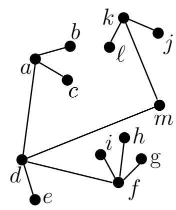
(a) Hypergraph formed by disks and points



(b) Dual-arrangement graph of disks



(c) Support on blue points for hypergraph in (b)



(d) Support for the hypergraph defined in (a)

Figure 1.4: Support for hypergraph defined by disks and points in the plane.

The motivation for adopting this graph-theoretic framework for hypergraphs may not be immediately evident. Below, we present three compelling reasons for pursuing this direction.

1. In the construction of a planar support for non-piercing regions in the plane [RR20], the authors used delicate topological arguments, however, our graph-theoretic approach is entirely combinatorial, employing only fundamental graph operations.
2. The non-piercing condition is insufficient for the existence of a sparse support for the dual hypergraph when the host graph G is defined on a surface of higher genus. See Chapter 2 for a concrete example. To overcome this, we defined a weaker notion of cross-free subgraphs, which generalize non-piercing in the plane.

3. The graph-based framework under consideration may have potential applications beyond the natural geometric setting.

§ 1.4 Preliminaries

In this section, we formally introduce the definitions and concepts that will be used throughout this thesis.

Definition 1.5 (Hypergraph)**.** A hypergraph is a pair $(V, \mathcal{E})$, where V is a set and $\mathcal{E}$ is a collection of subsets of V.

We call V the set of *vertices*, and $\mathcal{E}$ the set of *hyperedges* of the hypergraph.

Definition 1.6 (Support)**.** A support for a hypergraph $(V, \mathcal{E})$ is a graph Q on V such that $Q[E]$, the subgraph of Q induced by the vertices of E, is connected $\forall\, E \in \mathcal{E}$.

Given a hypergraph $(V, \mathcal{E})$, called the *primal hypergraph*, we define its *dual* hypergraph as a hypergraph that has a vertex for each element of $\mathcal{E}$ and hyperedge for each $v \in X$ defined by $\{E \in \mathcal{E} : v \in E\}$. Finally, given two collections $\mathcal{E}, \mathcal{K}$ of subsets of V, the *intersection* hypergraph is a hypergraph that has a vertex for each $E \in \mathcal{E}$, and a hyperedge for each $K \in \mathcal{K}$ defined by $\{E \in \mathcal{E} : K \cap E \neq \emptyset\}$. Intersection hypergraphs generalize both primal and dual hypergraphs: taking the elements in $\mathcal{E}$ as singleton subsets of V we get the primal, and taking the elements of $\mathcal{K}$ as singleton subsets of V we get the dual hypergraph. Now we are going to present the notion of a *support* that is central to this thesis.

This thesis primarily investigates hypergraphs that are derived from a fixed host graph, formally introduced as follows: Let $G = (V, E)$ be a given graph and $\mathcal{H}$ be a collection of subgraphs of G. This naturally defines a hypergraph on the vertex set V that has a hyperedge for each $H \in \mathcal{H}$ defined by the subset $V(H)$ of V.

Throughout this thesis, whenever we consider hypergraphs derived from a host graph G and a collection $\mathcal{H}$ of its subgraphs, we restrict our attention to the case where $\mathcal{H}$ comprises only *connected* subgraphs, reflecting the focus of this work. So, it is easy to see that for the primal hypergraph, a subgraph G' of G is a support, where G' is the subgraph of G induced on the vertices $\{V(H) : H \in \mathcal{H}\}$: indeed, by definition of $\mathcal{H}$, each hyperedge $H \in \mathcal{H}$, is a connected subgraph of G, hence it is connected subgraph of G'.

Consequently, to make the problem non-trivial and as motivated from the geometric setting discussed in Section 1.3.2, we restructure the notion of a primal hypergraph. Consider a partition of the vertices of the host graph G into *terminals* and *non-terminals*.

Then a connected subgraph of G need not remain connected when induced on its terminal vertices. Therefore, the goal will be to construct a sparse graph Q on the terminal vertices such that each subgraph $H \in \mathcal{H}$ induces a connected subgraph on its terminal vertices in Q. In the following, V always denotes the vertex set of G, and $V(X)$, the vertex set of a subgraph X of G. Mathematically, we describe the notion of a primal hypergraph and primal support as follows.

Definition 1.7 (Primal hypergraph)**.** Given a graph $G = (V, E)$, a 2-coloring (not necessarily proper) $c : V \to \{\mathbf{b}, \boldsymbol{r}\}$, and a collection $\mathcal{H}$ of connected subgraphs of G. The *primal hypergraph* defined by the pair $(G, \mathcal{H})$ is the hypergraph $(\mathbf{b}(V), \mathbf{b}(\mathcal{H}))$, where $\mathbf{b}(V) = \{v \in V : c(v) = \mathbf{b}\}$ and $\mathbf{b}(\mathcal{H}) = \{V(H) \cap \mathbf{b}(V) : H \in \mathcal{H}\}$.

We call $\mathbf{b}(V)$, the set of *blue* or terminal vertices, and $\boldsymbol{r}(V) = \{v \in V : c(v) = \boldsymbol{r}\}$, the set of *red* or non-terminal vertices of G.

Definition 1.8 (Primal support)**.** For a primal hypergraph $(\mathbf{b}(V), \mathbf{b}(\mathcal{H}))$, a *primal support* is a graph $Q = (\mathbf{b}(V), F)$ such that $\forall H \in \mathcal{H}$, the hyperedge $V(H) \cap \mathbf{b}(V)$ induces a connected subgraph of Q.

See Figs. 1.5a and 1.5d for a primal hypergraph and a primal support for it. For completeness and for the ease of exposition, we redefine below the concept of dual and intersection hypergraphs and their corresponding supports in the terminology of subgraphs of a host graph. For a family $\mathcal{H}$ of subgraphs of G, we write $\mathcal{H}_v = \{H \in \mathcal{H} : v \in V(H)\}$ for each vertex v of G. For two families $\mathcal{H}$ and $\mathcal{K}$ of subgraphs of G, we define $\mathcal{H}_K = \{H \in \mathcal{H} : V(H) \cap V(K) \neq \emptyset\}$. For $H \in \mathcal{H}$, we abuse notations, and use H to also denote the vertex corresponding to H in the dual and intersection hypergraphs.

Definition 1.9 (Dual hypergraph)**.** Let $G = (V, E)$ be a graph and $\mathcal{H}$ be a collection of connected subgraphs of G. The *dual hypergraph* is defined by the pair $(\mathcal{H}, \{\mathcal{H}_v\}_{v \in V})$.

In other words, the dual hypergraph has a vertex for each $H \in \mathcal{H}$ and a hyperedge $\mathcal{H}_v$ for each $v \in V$.

Definition 1.10 (Dual support)**.** For a dual hypergraph $(\mathcal{H}, \{\mathcal{H}\}_{v \in V})$, its support (called a *dual support*) is a graph $Q^* = (\mathcal{H}, F)$ on $\mathcal{H}$ such that $Q^*[\mathcal{H}_v]$, the subgraph of Q^* induced on $\mathcal{H}_v$, is connected $\forall v \in V$.

Figs. 1.5b and 1.5e show a dual hypergraph and a dual support.

Definition 1.11 (Intersection hypergraph)**.** Given two families $\mathcal{H}$ and $\mathcal{K}$ of connected subgraphs of G, the *intersection hypergraph* is a hypergraph $(\mathcal{H}, \{\mathcal{H}_K\}_{K\in\mathcal{K}})$.

More precisely, the intersection hypergraph has a vertex for each $H \in \mathcal{H}$, and a hyperedge $\mathcal{H}_K$ $\forall K \in \mathcal{K}$.

Definition 1.12 (Intersection support)**.** A support for the intersection hypergraph $(\mathcal{H}, \{\mathcal{H}_K\}_{K\in\mathcal{K}})$ (called an *intersection support*) is a graph $\tilde{Q} = (\mathcal{H}, F)$ on $\mathcal{H}$ such that for each $K \in \mathcal{K}$, the subgraph of $\tilde{Q}$ induced by the vertex set $\mathcal{H}_K$, is connected.

See Figs. 1.5c and 1.5f for an intersection hypergraph and an intersection support.

To make the definition symmetric, we could have considered a bi-coloring $c : \mathcal{H} \to \{\mathbf{b}, \boldsymbol{r}\}$, for the dual hypergraph and required that only $Q^*[\mathcal{H}_{\mathbf{b}_v}]$ be connected for each $v \in V$, where $\mathcal{H}_{\mathbf{b}_v} = \{H \in \mathcal{H}_v : c(H) = \mathbf{b}\}$. However, this reduces to the problem of constructing a dual support restricted to the subset of subgraphs $\{H \in \mathcal{H} : c(H) = \mathbf{b}\} \subseteq \mathcal{H}$, which is equivalent to what we considered in the definition of a dual support. An analogous argument holds for the intersection hypergraph as well. Therefore, it is sufficient to consider an uncolored version of the problems for the dual and intersection settings.

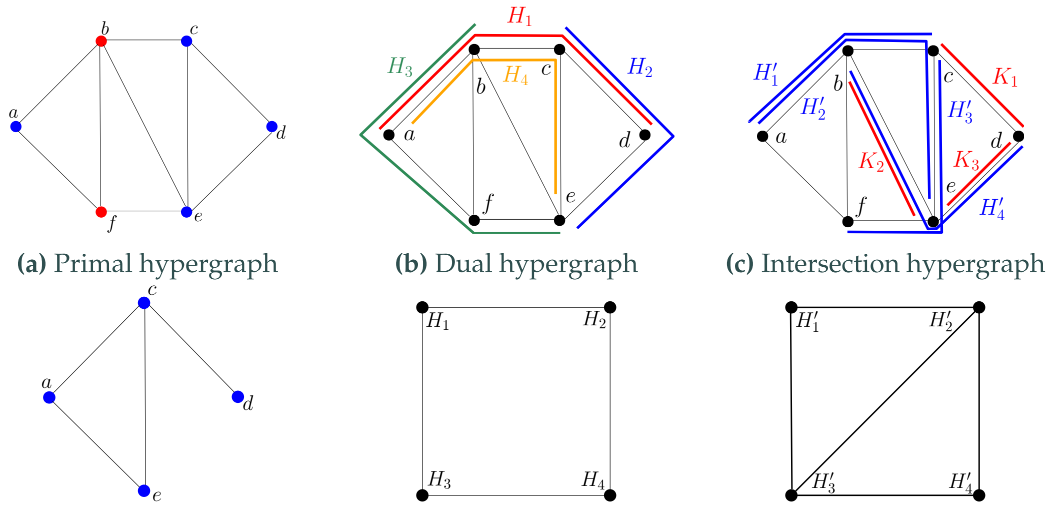
The intersection hypergraphs generalize primal and dual hypergraphs; the existence of an intersection support from a sparse family of graphs $\mathcal{G}$ also implies the existence of a primal and a dual support that lies in $\mathcal{G}$.

▷ Graph systems and Intersection systems.

Let G be a graph and $\mathcal{H}, \mathcal{K}$ be two families of connected subgraphs of G. We refer to the pair $(G, \mathcal{H})$ as a *graph system*, which serves as the underlying structure when analyzing a primal or a dual hypergraph. Likewise, we refer to the triple $(G, \mathcal{H}, \mathcal{K})$ as an *intersection system* used in the context of studying an intersection hypergraph. Therefore, whenever we use the terminology a primal or a dual hypergraph defined by a graph system $(G, \mathcal{H})$, we explicitly mean to consider the primal or dual hypergraphs as per Definitions 1.7 and 1.9 respectively. An analogous argument holds for an intersection hypergraph.

§ 1.5 Contributions

This section highlights our main contributions to this thesis. Recall from Section 1.2 that our motivation for the study of sparse supports comes from several optimization problems defined on hypergraphs, and as discussed, it is NP-hard to decide if an abstract hypergraph admits a support that is a 2-outerplanar graph. We investigate suf-



(a) Primal hypergraph **(b)** Dual hypergraph **(c)** Intersection hypergraph

(d) Primal support **(e)** Dual support **(f)** Intersection support

Figure 1.5: (a) and (b): Primal and Dual hypergraphs on the graph system $(G, \mathcal{H})$, where G is the graph on vertices $\{a, b \ldots, f\}$, and $\mathcal{H} = \{H_1, \ldots H_4\}$ with $H_1 = \{a, b, c, d\}$, $H_2 = \{c, d, e\}$, $H_3 = \{a, b, f, e\}$, $H_4 = \{a, b, c, e\}$; (c): Intersection hypergraph for $(G, \mathcal{H}', \mathcal{K})$ with $\mathcal{H}' = \{H'_1, \ldots, H'_4\}$ and $\mathcal{K} = \{K_1, \ldots, K_3\}$, where $H'_1 = \{a, b, c\}$, $H'_2 = \{a, b, c, e\}$, $H'_3 = \{c, e, f\}$, $H'_4 = \{b, d, e\}$, and $K_1 = \{c, d\}$, $K_2 = \{b, e\}$, $K_3 = \{d, e\}$. Figures in (d)-(f) show the respective support graphs.

ficient conditions under which the intersection hypergraph defined by a graph system admits a support that comes from a sparse family of graphs.

We primarily study two classes of graphs: (i) The class of graphs that are *2-cell embedded* on an orientable surface of *genus* g for some constant g, henceforth called *embedded graphs*. For a formal description of these notions, Definitions 1.13 and 1.14. (ii) The class of graphs that have bounded *treewidth* (Chapter 4). As an intermediate case, we also study the setting where G is an outerplanar graph (Chapter 3).

While the results for bounded genus graphs roughly follow the proof outline of [RR20], several new ideas are required for the proofs to go through. In particular, it turns out that for graphs of bounded genus, the non-piercing condition is insufficient for the existence of sparse supports. We introduce the notion of *cross-free* subgraphs (see Definition 2.4 in Chapter 2). If $\mathcal{H}$ and $\mathcal{K}$ are two collections of cross-free subgraphs of G, and G has genus g, then graph systems $(G, \mathcal{H})$ and $(G, \mathcal{H}, \mathcal{K})$ are called *cross-free systems* of genus g. Similarly, if $\mathcal{H}$ and $\mathcal{K}$ are collections of non-piercing subgraphs of G, and G has treewidth t, then $(G, \mathcal{H})$ and $(G, \mathcal{H}, \mathcal{K})$ are called *non-piercing systems* of treewidth t.

Now we are ready to state our main results. Following the results for the construction of supports for hypergraphs derived from a host graph, we state our results for hypergraphs defined by points and axis-aligned non-piercing rectangles in the plane. Then we state some applications, followed by some hardness results on abstract hypergraphs.

§ 1.5.1 Construction of sparse support graphs

In this section, we state our results for the construction of primal, dual, and intersection supports for hypergraphs defined on a given host graph G. We explicitly assume that for all the results stated for a primal support, an arbitrary 2-coloring (not necessarily proper) $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ of the vertices of G is given as input, and the goal to is construct a support on the vertices $\mathbf{b}(V)$, where V is the vertex set of G.

We also state our results for the construction of a planar support for geometric hypergraphs defined by points and non-piercing rectangles in the plane.

Support of bounded genus

We start with the definitions of *embedding* and *genus* of a graph.

Definition 1.13 (Embedding of a graph)**.** A graph G is said to be *embedded* on a surface Σ if the vertices of G are distinct points on Σ and each edge of G is a simple arc lying in Σ whose endpoints are the vertices of the edge such that its interior is disjoint from other edges and vertices. A *2-cell embedding* is an embedding of a graph on a surface, where each face is homeomorphic to a disk in the plane.

We say that a graph G has an *embedding* on a surface Σ if there is a graph G' embedded in Σ such that G' is isomorphic to G.

Definition 1.14 (Genus)**.** An orientable surface has genus g if it is obtained from a sphere by adding g *handles* to it. The *genus* g of a graph G is the minimum genus of an oriented surface Σ so that G has an embedding in Σ.

We refer the reader to Chapter 3 of [MT01] for more details on graphs embedded on surfaces. We proved the following results for embedded graphs.

Theorem 1.2. Let $(G, \mathcal{H})$ be a cross-free system of genus g. Then, there exists a primal support Q on $\mathbf{b}(V)$ of genus at most g.

Theorem 1.3. If $(G, \mathcal{H})$ is a cross-free system of genus g, then there exists a dual support Q^* on $\mathcal{H}$ of genus at most g.

Theorem 1.4. If $(G, \mathcal{H}, \mathcal{K})$ is a cross-free intersection system of genus g, there exists an intersection support $\tilde{Q}$ on $\mathcal{H}$ of genus at most g. In other words, for each $K \in \mathcal{K}$, the induced subgraph $\tilde{Q}[\mathcal{H}_K]$ of Q is connected.

These results directly generalize the results of [RR20] since the cross-free condition generalizes the non-piercing in the plane. The techniques used in the construction of supports have the advantage of using basic graph operations and not requiring intricate topological arguments.

Outerplanar support

Let's first define an outerplanar graph.

Definition 1.15 (Outerplanar graph)**.** A graph G is called outerplanar if there is an embedding of G in the plane such that all its vertices lie on the exterior face.

Let G be an outerplanar graph, and $\mathcal{H}$ and $\mathcal{K}$ be two collections of non-piercing subgraphs of G. Then, we show the following.

Theorem 1.5. For any 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$, there is an outerplanar support for the primal hypergraph defined by the pair $(G, \mathcal{H})$, which can be constructed in $O(n^6)$-time.

Theorem 1.6. There is an outerplanar support for the dual hypergraph defined by $(G, \mathcal{H})$, which can be constructed in $O(n^6)$-time.

Theorem 1.7. There is an outerplanar support for the intersection hypergraph defined by $(G, \mathcal{H}, \mathcal{K})$, which can be constructed in $O(n^6)$-time.

In each of the aforementioned results, an appropriate support can be computed in time polynomial in n, where n is the number of vertices in G. Our analysis relies on the pivotal notion of *Vapnik-Chervonenkis dimension (VC-dimension)* (Definition 3.5) and the Sauer-Shelah lemma (Lemma 3.4) that bounds the quantities $|\mathcal{H}|$ and $|\mathcal{K}|$.

We will see in Chapter 3 that for the construction of an outerplanar primal or dual support, some weaker conditions are sufficient. For the sake of precision, we refrain from stating those results here.

Support of bounded treewidth

We first define *tree decomposition*, and *treewidth* of a graph, introduced by Robertson and Seymour [RS86].

Definition 1.16 (Tree decomposition)**.** Given a graph $G = (V, E)$, a tree decomposition of G is a pair $(T, \mathcal{B})$, where T is a tree and $\mathcal{B}$ is a collection of *bags* - subgraphs of G indexed by the nodes of T, that satisfies the following properties:

1. For each $v \in V(G)$, the set of bags of T containing v induces a sub-tree of T.
2. For every edge $\{u, v\}$ in G, there is a bag $B \in \mathcal{B}$ such that $u, v \in B$.

Definition 1.17 (Treewidth)**.** The width of a tree decomposition $(T, \mathcal{B})$ is defined to be $\max_{x \in V(T)} |B_x| - 1$. The treewidth of a graph G is the minimum width over all the tree decompositions of G, and is denoted $\textsc{tw}(G)$.

We use $\textsc{tw}(H)$ to denote the treewidth of a graph H. In the following, let $\textsc{tw}(G) = t$, and n is the number of vertices in G.

Theorem 1.8 (Informal)**.** Let $(G, \mathcal{H})$ be a non-piercing system of treewidth t. Then, there is a primal support Q such that $\textsc{tw}(Q) = O(2^t)$, which can be computed in time poly(n^t).

Theorem 1.9 (Informal)**.** Let $(G, \mathcal{H})$ be a non-piercing system of treewidth t. There is a dual support Q^* such that $\textsc{tw}(Q) = O(2^{4t})$, which can be computed in time poly(n^t).

Theorem 1.10 (Informal)**.** Let $(G, \mathcal{H}, \mathcal{K})$ be a non-piercing intersection system of treewidth t. There exists an intersection support $\tilde{Q}$ such that $\textsc{tw}(\tilde{Q}) = 2^{O(2^t)}$. Further, such an intersection support can be computed in time poly(n^{2^t}).

The results above show that our algorithms to construct a primal, dual, or intersection support are FPT in $\textsc{tw}(G)$. In other words, if the parameter t is bounded above by a constant, then our algorithms run in polynomial time. We again make use of VC-dimension and the Sauer-Shelah lemma to establish the claimed running time.

The exponential blow-up in the treewidth of primal and dual support is sometimes necessary, as the following results show.

Theorem 1.11 (Informal)**.** There is a non-piercing graph system $(G, \mathcal{H})$ of treewidth t and a 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ such that $\text{TW}(Q) = O(2^t)$ for any primal support Q.

Theorem 1.12 (Informal)**.** There is a non-piercing graph system $(G, \mathcal{H})$ of treewidth t such that $\text{TW}(Q^*) = O(2^t)$ for any dual support Q^*.

Planar support for non-piercing rectangles

Now consider hypergraphs arising from geometric configurations in the Euclidean plane. Let $(P, \mathcal{R})$ be a hypergraph, where P is a set of n points, and $\mathcal{R}$ is a set of m axis-parallel rectangles in the plane, where each $R \in \mathcal{R}$ defines a hyperedge consisting of points in $P \cap R$. The goal is to construct a graph Q on P such that for each $R \in \mathcal{R}$, the vertices $P \cap R$ induce a connected subgraph of Q. Note that, in this context, we do not need to consider a 2-coloring of the vertices of the hypergraph. We show the following:

Theorem 1.13. If $\mathcal{R}$ is a set of non-piercing rectangles, then $(P, \mathcal{R})$ admits a planar support with straight line embedding that can be computed in time $O(n \log^2 n + (n + m) \log m)$.

Two rectangles R, R' are said to be *piercing* if they are *not* non-piercing, i.e., if $R \setminus R'$ (and hence $R' \setminus R$) is disconnected.

Theorem 1.14. If for each $p \in \mathbb{R}^2$, p is contained in at most $k - 1$ rectangles of $\mathcal{R}$ that are pairwise piercing, then there is a support Q for $(P, \mathcal{R})$ such that Q is a union if at most k planar graphs.

Our result for the construction of a planar support seems very restrictive since Raman and Ray [RR20] gave a polynomial time algorithm for any collection of non-piercing regions in the plane. However, our algorithm is robust in a 3-fold way: it is simple, significantly faster than the algorithm in [RR20] that computes a planar support in time $O(m^2(\min\{m^3, mn\} + n))$, and finally, our algorithm computes a planar support with a straight-line embedding of the graph. We believe that the tools we developed in this construction could potentially be applied with some modifications, to construct a sparse support for other hypergraphs that arise from objects with other geometric properties, e.g., convex objects or *homothets*[6] of convex objects in $\mathbb{R}^2$ or higher dimensions.

[6] For a convex object C in the plane, a homothet of C is an object C', where C' can obtained from C by uniform scaling and translation operations, i.e., we are not allowed to rotate the objects.

§ 1.5.2 Applications of support

Now we state some algorithmic applications of support for packing, covering, and coloring problems defined on hypergraphs. Before we proceed, we introduce two notions that are variants of the notion of support.

For an abstract hypergraph $(V, \mathcal{E})$, a *weak support* is a graph Q on V such that for each hyperedge E containing at least two vertices, there is an edge in Q with both end-points in E. Further, let $c : V \to \{\mathbf{b}, \boldsymbol{r}\}$ be any two coloring of V. Then a *weak bipartite support* for $(V, \mathcal{E})$ is a bipartite graph Q' on V with bipartition $\boldsymbol{r}(V)$ and $\mathbf{b}(V)$ such that for each $E \in \mathcal{E}$ that contains a vertex in $\boldsymbol{r}(V)$ and $\mathbf{b}(V)$, there is an edge in Q' with both end-points in E. Note that if Q is a support, it is also a weak support and a weak bipartite support.

§ 1.5.2.1 PTAS for Packing and Covering Problems

As stated in Section 1.2, the motivation of our study on the existence of support comes from better approximation algorithms for optimization problems on hypergraphs. For several packing and covering problems, a simple local search algorithm yields a PTAS. We only give a short description of how the local search algorithm works. We will discuss it in more detail in Chapter 6.

For an optimization problem Π, let $\mathcal{O}$ be an optimal solution, and $\mathcal{L}$ be a solution returned by some local search algorithm. We require a local search graph on $\mathcal{L} \cup \mathcal{O}$, i.e., a graph that satisfies the local search property and has a sublinear-sized balanced separator. If Q is a support for the hypergraph under consideration, then the first property follows from the fact that each hyperedge induces a connected subgraph of Q. The second requirement is satisfied if the support comes from a hereditary family of graphs with sublinear-sized separators. Since graphs of bounded genus, outerplanar graphs and bounded treewidth graphs all have sublinear-sized separators [GHT84; AST90], our results for the existence of appropriate supports, combined with the framework in [CH12; MR10; BR+18; RR20] directly imply the following results. These results generalize the results in [RR20] in the plane, as well as on oriented surfaces of higher genus.

A collection $\mathcal{D}$ of connected regions on an oriented surface Σ is said to be *weakly non-piercing* if for each $D, D' \in \mathcal{D}$, either $D \setminus D'$ or $D' \setminus D$ is connected. Note that the regions need not be in general position, i.e., the boundaries of any two regions need not intersect finitely, nor do they require crossing at the point of intersection. In addition, more than two regions are allowed to share a common point on their boundaries. This notion is weaker than the non-piercing condition considered by Raman and Ray

[RR20], where both $D \setminus D'$ and $D' \setminus D$ are required to be connected, and the regions are in general position.

Given two collections $\mathcal{H}$ and $\mathcal{K}$ of connected regions on a surface, the *intersection hypergraph of regions* takes a vertex for each $H \in \mathcal{H}$, and a hyperedge for each $K \in \mathcal{K}$ consisting of all $H \in \mathcal{H}$ that intersect K (see Definition 6.19). If $\mathcal{H}$ and $\mathcal{K}$ are families of simply-connected weakly non-piercing regions on a bounded genus surface, we obtain PTAS for packing and covering problems defined by the corresponding geometric hypergraphs. We refer the reader Section 6.1 for the definitions of packing and covering problems that we use in the results stated in this section.

In the following results, Σ is an oriented surface of bounded genus. Let $\mathcal{H}$ and $\mathcal{K}$ be two sets of simply-connected weakly non-piercing regions on Σ. For the informal statements below, we write "regions" to mean "simply-connected weakly non-piercing regions".

Theorem 1.15 (Informal)**.** Given families $\mathcal{H}$ and $\mathcal{K}$ of regions on Σ. There is an embedded cross-free intersection system $(G, \mathcal{H}', \mathcal{K}')$ on Σ for some graph G such that the corresponding intersection hypergraph is isomorphic to the intersection hypergraph defined by $\mathcal{H}$ and $\mathcal{K}$.

Theorems 1.4 and 1.15 along with the framework of [CH12; MR10; BR+18; RR20] implies the following results for packing and covering problems. These results hold for any cross-free system defined by subgraphs of an embedded graph; however, we state here only for the regions on a surface.

Theorem 1.16 (Informal)**.** Given families $\mathcal{H}, \mathcal{K}$ of regions on Σ, and a non-negative capacity function c on $\mathcal{H}$. Then the Generalized Capacitated Packing problem admits a PTAS.

This implies the following.

Corollary 1.17 (Informal)**.** The capacitated Set Packing/Point Packing problem defined by a point set P and regions $\mathcal{H}$ on Σ admits a PTAS.

Theorem 1.18 (Informal)**.** Given families $\mathcal{H}, \mathcal{K}$ of regions on Σ. The Generalized Covering problem admits a PTAS.

This implies the following.

Corollary 1.19 (Informal)**.** The Set Cover/Hitting Set problem defined by a point set P and regions $\mathcal{H}$ on Σ admits a PTAS.

Theorem 1.20 (Informal)**.** The Dominating Set problem for the *intersection graph* of regions $\mathcal{H}$ on Σ admits a PTAS.

If the capacities are unit capacities, then we need only a weak bipartite support for an appropriate primal, dual or intersection hypergraph to ensure a PTAS for the problems above. However, if the capacities are not unit capacities, then we need a support for the analysis of the local search algorithm. We refer the reader to Table 6.1 that summarises the hypergraph and the variation of support required for each of the problems considered above.

Further, the requirement that the regions are simply connected is essential, as [RR20] show an example of weakly non-piercing and connected regions whose support is $K_{r,r}$ for any $r \in \mathbb{N}$, and hence do not have sublinear-sized separators. We also show the existence of non-piercing graph systems that are *not* cross-free, and the Set Cover and Hitting Set problems are APX-hard for hypergraphs defined on such systems.

By the existence of primal and dual supports, we also obtain similar results for non-piercing systems of bounded treewidth. For instance, we can give a PTAS for packing and covering problems defined on cliques in a graph.

The next result shows a constant-factor approximation algorithm for the Hitting Set problem on axis-aligned rectangles in the plane.

Theorem 1.21 (Informal)**.** Let P be a point set and $\mathcal{R}$ be a collection of axis-parallel rectangles in $\mathbb{R}^2$ such that any point of $\mathbb{R}^2$ lies in at most k rectangles that are pairwise *piercing*. Then there is a $(k + \epsilon)$-approximation algorithm for the Hitting Set problem on $(P, \mathcal{R})$ for any $\epsilon > 0$.

Remark 1.5.1. For all the algorithmic results above, what we need is the *existence* of sparse supports, which are precisely the graphs from a hereditary class with sublinear-sized balanced separators [DN16]. Sparsity is a general requirement; however, in this thesis, we focus on some special classes of sparse graphs. In particular, we investigate the conditions under which the associated hypergraph admits a support that is (outer)planar, has bounded genus or bounded treewidth.

§ 1.5.2.2 Hypergraph coloring

Besides the results for the algorithmic problems of packing and covering, our results also have applications in the context of hypergraph coloring. A *proper coloring* of a hypergraph is a coloring of its vertices so that no hyperedge with at least two vertices is *monochromatic*[7]. A hypergraph is *k-colorable* if there is a proper coloring with k colors. If Q is a weak support for a hypergraph $\mathcal{H} = (V, \mathcal{E})$, then for any $E \in \mathcal{E}$ containing at least two vertices, there exist $u, v \in E$ s.t. u and v are adjacent in Q, by definition. Therefore if Q is k-colorable, so is $\mathcal{H}$.

Keller and Smorodinsky [KS18] showed that the intersection hypergraph of disks in the plane is 4-colorable, i.e., a coloring with 4 colors such that no hyperedge is monochromatic (assuming that the size of each hyperedge is at least two). This was generalized by Keszegh [Kes20] for pseudodisks, which was further generalized by Raman and Ray in [RR20] to show that the intersection hypergraph of non-piercing regions is 4-colorable since the underlying hypergraph admits a planar support. Our results generalize all these existing results to hypergraphs defined by regions on higher genus surfaces.

Let P be a set of points, and $\mathcal{H}$ and $\mathcal{K}$ be two collections of simply connected weakly non-piercing regions on an oriented surface Σ of genus g. Leveraging the cross-free property of an appropriate graph system induced by these regions together with the existence of appropriate primal, dual, and intersection supports of genus at most g, we obtain the following result.

Theorem 1.22 (Informal)**.** For the primal hypergraph $(P, \mathcal{H})$, there is a coloring of the points of P with k_g colors such that each region that contains more than one point, contains points of at least 2 colors.

Theorem 1.23 (Informal)**.** For the dual hypergraph $(\mathcal{H}, P)$, there is a coloring of the regions in $\mathcal{H}$ with k_g colors such that every point that is covered by more than one region, is covered by regions of at least 2 colors.

Theorem 1.24 (Informal)**.** For the intersection hypergraph $(\mathcal{H}, \mathcal{K})$, there is a coloring of $\mathcal{H}$ with k_g colors so that for any $K \in \mathcal{K}$, the set of regions in $\mathcal{H}$ intersecting K, is not monochromatic.

Here, $k_g = \left(\frac{7+\sqrt{1+48g}}{2}\right)$ is an upper bound on the chromatic number of a graph of genus g [RY68; MT01].

[7] A hyperedge is monochromatic if all its vertices get the same color.

Ackerman et al. [AKP20] studied $ABAB$-free hypergraphs, which we will define in the upcoming subsection. The authors showed that these hypergraphs are 3-colorable. We discuss the connection of their result with ours in Chapter 6.

Apart from a proper coloring of hypergraphs, the notion of *conflict-free*[8] coloring, introduced by Even et al. [Eve+03], is also well studied in the past two decades. Since for constant g the hypergraphs we considered can be colored with k_g colors, using the general framework in [Eve+03], it can be shown that $O(\log n)$ (with n being the number of vertices) colors suffice for a conflict-free coloring, when the underlying hypergraphs come from a hereditary class. Our results, in particular Corollary 6.15 generalize the earlier work on geometric hypergraphs [Eve+03; KS18; Kes20; RR20].

§ 1.5.3 NP-hardness of $ABAB$-free recognition

Given an abstract hypergraph $\mathcal{H} = (V, \mathcal{E})$ and a linear ordering π of the vertices V. We say that two hyperedges $H, L \in \mathcal{E}$ *form an* $ABAB$*-pattern in* π if there are distinct vertices $h_1, l_1, h_2, l_2 \in V$ in this order in π such that $h_1, h_2 \in H \setminus L$, and $l_1, l_2 \in L \setminus H$. We say that π is an $abab$-free ordering for $\mathcal{H}$ if there is no pair of hyperedges H, L that form an $ABAB$-pattern in π. $\mathcal{H}$ is said to be $ABAB$-free if there is an ordering of V that is $ABAB$-free. The notion can be extended to $ABABA$-free hypergraphs. That is, $\mathcal{H}$ is $ABABA$-free if there is an ordering π of V such that for any pair of hyperedges H and L, there are no five vertices h_1, l_1, h_2, l_2, h_3 in this order in π such that $h_1, h_2, h_3 \in H \setminus L$, and $l_1, l_2, \in L \setminus H$.

In fact, the notion can be generalized to $(AB)^k$-free and $(AB)^k A$-free hypergraphs for any $k \in \mathbb{N}$, where for $k = 2$, $(AB)^2$-free is same as $ABAB$-free, and $(AB)^2 A$-free is $ABABA$-free. The explicit generalized definitions can be found in Chapter 7. We proved the following results in this context:

> **Theorem 1.25.** For any positive integer $k \geq 2$, it is NP-complete to decide if a given hypergraph is $(AB)^k$-free.

> **Theorem 1.26.** For any positive integer $k \geq 2$, it is NP-complete to decide if a given hypergraph is $(AB)^k A$-free.

This abstract description of hypergraphs by forbidden patterns of sequences has a strong connection with their geometric representations. We say that a hypergraph $\mathcal{H} = (V, \mathcal{E})$ is *realizable* by a collection $\mathcal{R}$ of regions in the plane if there is a point set

[8] Conflict-free coloring of a hypergraph is a proper coloring, where each hyperedge contains a uniquely colored vertex.

$P \subseteq \mathbb{R}^2$, and bijections $\phi_1 : V \to P$ and $\phi_2 : \mathcal{E} \to \mathcal{R}$ such that for each for any $v \in V$ and $E \in \mathcal{E}$, the vertex v lies in E if and only if $\phi_1(v)$ lies in $\phi_2(E)$. In other words, these mappings preserve the incidences.

Ackerman et al. [AKP20] showed that $ABAB$-free hypergraphs are exactly the hypergraphs that can be realized by *stabbed*[9] pseudodisks in the plane. Pseudodisks are 2-intersecting curves, i.e., for any two pseudodisks, their boundaries cross each other at most twice. The authors also generalized the result to the realization of $(AB)^k$-free and $(AB)^kA$-free hypergraphs by *t-intersecting upward curves*[10] for $t = 2k - 2$ and $t = 2k - 1$, respectively.

Our results then imply the NP-hardness for the above realization of a hypergraphs by t-intersecting upward curves for any $t \geq 2$. With a slight modification of our result for stabbed pseudodisks, we show that it is NP-hard to decide if a hypergraph can be realized by pseudodisks in the plane.

§ 1.6 Remarks on Notation and Conventions

We recap the following notations that will be used throughout this thesis.

For any graph G, we use $V(G)$ and $E(G)$ to denote, respectively, the set of vertices and the set of edges of G. We also use V rather than $V(G)$ if G is clear from the context. Given a vertex v, we also use $v \in G$ to mean $v \in V(G)$. Similarly, for an edge e, we write $e \in G$ to mean $e \in E(G)$. For a subset $S \subseteq V(G)$, $G[S]$ denotes the subgraph of G induced on the vertex set S.

All graphs and hypergraphs considered in this thesis are finite. All graphs are simple unless otherwise stated.

Let G be a given host graph, and let $\mathcal{H}$ and $\mathcal{K}$ be two collections of subgraphs of G. Throughout this thesis, we consider the following types of hypergraphs derived from these objects:

Primal and Dual hypergraphs associated with the pair $(G, \mathcal{H})$.
Intersection hypergraphs associated with the triple $(G, \mathcal{H}, \mathcal{K})$.

Unless stated otherwise, the following assumptions hold uniformly across all such

[9] An arrangement of pseudodisks in $\mathbb{R}^2$ is said to be *stabbed* if there is a point $p \in \mathbb{R}^2$ such that p lies in all the pseudodisks of the arrangement.

[10] t-intersecting upward curves are the regions in $\mathbb{R}^2$ each having its boundary defined by an x-monotone curve, and boundaries of any two of them intersect at most t times. Further, the region enclosed by a curve is defined by the pseudo halfplane above the curve.

hypergraph constructions considered in this thesis.

1. The subgraphs in $\mathcal{H}$ and $\mathcal{K}$ are connected subgraph G.

2. If the subgraphs in $\mathcal{H}$ and $\mathcal{K}$ are replaced by the corresponding *induced* subgraphs in G, the underlying primal, dual and intersection hypergraphs remain the same. Moreover, properties such as being *cross-free* or *non-piercing* are preserved under this replacement. Hence, we assume throughout that subgraphs in $\mathcal{H}$ and $\mathcal{K}$ are induced.

3. For a primal hypergraph defined by $(G, \mathcal{H})$, we use $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ to denote an arbitrary 2-coloring of the vertices of G, which need not be a proper coloring. A vertex colored $\mathbf{b}$ (respectively $\boldsymbol{r}$) refers to a *blue* (respectively *red*) vertex.

4. For any set $X \subseteq V(G)$, we denote $\mathbf{b}(X) = \{x \in X : c(x) = \mathbf{b}\}$, and $\boldsymbol{r}(X) = \{x \in X : c(x) = \boldsymbol{r}\}$. For a subgraph H of G, $\mathbf{b}(H)$ (respectively $\boldsymbol{r}(H)$) denote the set of blue (respectively red) vertices of H.

5. Since we will be dealing with induced subgraphs, we make use of the set-theoretic operations like intersection, union or set differences for the subgraphs at hand. Hence, for any two subgraphs X, Y of G, $X \cap Y$ refers to the subgraph induced on $V(X) \cap V(Y)$. The notations $X \cup Y$ and $X \setminus Y$ will be used analogously. We also use $X \subseteq Y$ to mean $V(X) \subseteq V(Y)$.

6. For a vertex v and an edge e of G, $\mathcal{H}_v = \{H \in \mathcal{H} : v \in V(H)\}$ and $\mathcal{H}_e = \{H \in \mathcal{H} : e \in (H)\}$. The terms $\mathcal{K}_v$ and $\mathcal{K}_e$ are defined analogously.

7. While considering a primal hypergraph, the term *support* refers to a *primal support*. A similar statement holds for dual and intersection hypergraphs.

8. Since for a dual or intersection support Q, the vertex set of Q is the elements of $\mathcal{H}$, for each $H \in \mathcal{H}$, we also use the notation H to denote the corresponding vertex in Q, for the ease of exposition.

▷ A guide to the reader.

We start with Chapter 2 for the construction of support of bounded genus for cross-free graph systems. In Chapters 3 and 4, we construct outerplanar and bounded treewidth supports for non-piercing graph systems. In Chapter 5, we show planar support for rectangles, followed by describing several applications of our results in Chapter 6 on the existence and construction of supports. We show our hardness results in Chapter 7, and conclude in Chapter 8 with some open problems and future research directions.

(A) Construction of Supports

Chapter 2

Support of Bounded Genus

Abstract

In this chapter, we show that if G is a graph of genus g with $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$, a two coloring of $V(G)$, and $\mathcal{H}, \mathcal{K}$ are two families of cross-free subgraphs of G, then the graph system $(G, \mathcal{H})$ admits a primal and a dual support of genus at most g, and the intersection system $(G, \mathcal{H}, \mathcal{K})$ admits an intersection support of genus at most g. The results in this chapter generalize the results of Raman and Ray [RR20] in the plane to higher genus surfaces. The material presented in this chapter is derived from the following publication.
"**On Supports for graphs of bounded genus**".

Table of Contents

§ 2.1 Introduction

Given a host graph $G = (V, E)$ and a partition of its vertices into two sets: a set of *terminals* and a set of *non-terminals*. Formally, we define a 2-coloring (not necessarily a proper coloring) of $V(G)$ for such a partition, namely, $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. The set $\mathbf{b}(V) = c^{-1}(\mathbf{b})$ is called the set of *blue vertices*, and $\boldsymbol{r}(V) = c^{-1}(\boldsymbol{r})$ is called the set of *red vertices*.

Let $\mathcal{H}, \mathcal{K}$ be two collections of subgraphs of G. We repeat some terminology from Chapter 1 for completeness. We call the tuple $(G, \mathcal{H})$ a *graph system*, and it defines a primal and a dual hypergraph. Likewise, we call the triple $(G, \mathcal{H}, \mathcal{K})$ an *intersection system* that defines an intersection hypergraph. The intersection hypergraph generalizes both primal and dual hypergraphs.

Let $\mathcal{H}, \mathcal{K}$ be two collections of subgraphs of G that satisfy a topological property of being *cross-free* (see Definition 2.4). We give algorithms for the construction of a primal and a dual support for the cross-free graph system $(G, \mathcal{H})$, and an intersection support for the cross-free intersection system $(G, \mathcal{H}, \mathcal{K})$. However, we are unable to show the efficiency of the running time of our algorithms. We contrast non-piercing with cross-free in the plane and show that non-piercing implies cross-free in the plane; however, the two notions are incomparable in higher genus surfaces.

▷ Organization of the chapter.

We start with defining some useful notions in Section 2.2. In Section 2.3, we contrast non-piercing with cross-free on orientable surfaces. In Section 2.4, we define the main technical operation- *vertex bypassing*, that we will use to construct the required supports. In Section 2.5, we give algorithms for the construction of primal, dual, and intersection supports, and we conclude in Section 2.6 with some open questions.

§ 2.2 Preliminaries

We recall the definitions of *embedding* and *genus* of a graph, from Chapter 1.

> **Definition 2.1** (Embedding of a graph)**.** A graph G is said to be *embedded* on a surface Σ if the vertices of G are distinct points on Σ and each edge of G is a simple arc lying in Σ whose endpoints are the vertices of the edge such that its interior is disjoint from other edges and vertices. A *2-cell embedding* is an embedding of a graph on a surface, where each face is homeomorphic to a disk in the plane.

A graph G has an *embedding* on a surface Σ if there is a graph G' embedded in Σ such that G' is isomorphic to G.

Definition 2.2 (Genus)**.** An orientable surface has genus g if it is obtained from a sphere by adding g *handles* to it. The *genus* g of a graph G is the minimum genus of an oriented surface Σ so that G has an embedding in Σ.

For more details on the above notions, see Chapter 3 of [MT01]. We say that a graph has bounded genus if it can be embedded on a surface whose genus is bounded. It should be noted that contracting any edge of a graph does not increase the genus of the resulting graph, and we will use this fact subsequently in this chapter.

Recall, as mentioned in previous chapters, that it is NP-hard to decide if a hypergraph admits a support that is planar [JP87] or even if it is 2-outerplanar [Buc+11]. Our goal is to consider restrictions on the hyperedges of the hypergraph so that the support is guaranteed to have bounded genus. To that end, we introduce the notion of cross-free hypergraphs and non-piercing hypergraphs. We start with the definition of a reduced graph.

Definition 2.3 (Reduced graph)**.** Let $(G, \mathcal{H})$ be a graph system. For two subgraphs $H, H' \in \mathcal{H}$, the *reduced graph* $R_G(H, H')$ (or just $R(H, H')$ if G is clear from context) is the graph obtained from G by contracting all edges, both of whose end-points are in $H \cap H'$.

Note that if G is embedded on a surface Σ, then this induces an embedding of $R_G(H, H')$ in Σ.

Definition 2.4 (Cross-free at v)**.** Let $(G, \mathcal{H})$ be an embedded graph system. Two subgraphs $H, H' \in \mathcal{H}$ are said to be cross-free at $v \in V(G)$ if the following holds: Consider the induced embedding of the reduced graph $R(H, H')$. Let $\tilde{v}$ be the image of v in $R(H, H')$. There are no 4 edges $e_i = \{\tilde{v}, v_i\}$ in $R(H, H')$, $i = 1, \ldots, 4$ incident to $\tilde{v}$ in cyclic order around $\tilde{v}$, such that $v_1, v_3 \in H \setminus H'$, and $v_2, v_4 \in H' \setminus H$.

For an embedded graph system $(G, \mathcal{H})$, if every pair $H, H' \in \mathcal{H}_v$ is cross-free at v, then $(G, \mathcal{H})$ is said to be cross-free at v. For two subgraphs H, H' if v is not contained in both H and H', then H and H' are assumed to be cross-free at v. The embedded graph system $(G, \mathcal{H})$ is cross-free if it is cross-free at all $v \in V(G)$. Finally, a graph system $(G, \mathcal{H})$ is cross-free if there exists an embedding of G such that the embedded graph system $(G, \mathcal{H})$ is cross-free with respect to $\mathcal{H}$. If there exist $H, H' \in \mathcal{H}$ such that H and H' are not cross-free at v, we say that H and H' are *crossing* at v.

An intersection hypergraph defined by the intersection system $(G, \mathcal{H}, \mathcal{K})$ is cross-free if there is an embedding of G such that the embedded graph systems $(G, \mathcal{H})$ and $(G, \mathcal{K})$ are simultaneously cross-free. Note that we can have $H \in \mathcal{H}$ and $K \in \mathcal{K}$ such that H and K are crossing.

Note 2.2.1. Throughout this chapter, we explicitly assume that G has genus g and it is 2-cell embedded on an orientable surface of genus g such that the embedding is cross-free with respect to $\mathcal{H}$ and $\mathcal{K}$, i.e., both $(G, \mathcal{H})$ and $(G, \mathcal{K})$ are simultaneously cross-free. We use the term $(G, \mathcal{H})$ is a cross-free system of genus g or the term $(G, \mathcal{H}, \mathcal{K})$ is a cross-free intersection system of genus g to mean that $(G, \mathcal{H})$ and $(G, \mathcal{K})$ are cross-free, and the host graph G has genus g.

Definition 2.5 (Non-piercing subgraphs)**.** Let $\mathcal{H}$ be a collection of connected induced subgraphs of G. two subgraphs $H, H' \in \mathcal{H}$ are called *non-piercing*, if both $H \setminus H'$ and $H' \setminus H$ induce connected subgraphs of G. $\mathcal{H}$ is called non-piercing if H and H' are non-piercing for all $H, H' \in \mathcal{H}$.

If $\exists\ H, H' \in \mathcal{H}$ such that either the induced subgraph $H \setminus H'$ of H or the induced subgraph $H' \setminus H$ of H' is not connected, then we say that H and H' are *piercing*.

Note that non-piercing is a purely combinatorial notion, and unlike the cross-free property, it does not require an embedding of the graph. Recall item 2 of Section 1.6 that we explicitly assume the subgraphs in $\mathcal{H} \cup \mathcal{K}$ to be connected induced subgraphs of G.

§ 2.3 Non-piercing and Cross-free Systems

The non-piercing condition implies the cross-free condition in the plane, but they are incomparable in higher genus surfaces. We start with the following result that shows that if a graph system is non-piercing in the plane, it is cross-free.

Theorem 2.1. Let $(G, \mathcal{H})$ be a non-piercing graph system, where G is a planar graph. Then, $(G, \mathcal{H})$ is cross-free.

Proof. We show that if $(G, \mathcal{H})$ is not cross-free, then it cannot be non-piercing. Let there be two subgraphs $H, H' \in \mathcal{H}$, a vertex x in $R_G(H, H')$ that lies in $H \cap H'$, and four edges incident to x with other end points $x_1, \ldots, x_4$ in cyclic order around x such that $x_1, x_3 \in H \setminus H'$ and $x_2, x_4 \in H' \setminus H$.

Since $\mathcal{H}$ is non-piercing, H is connected and $H \setminus H'$ induces a connected subgraph of G. Further, note that H and H' are non-piercing in G, then they remain non-piercing in $R_G(H, H')$ since $R_G(H, H')$ is obtained by contracting edges with both end points in $H \cap H'$. Therefore, there is an x_1-x_3 path P in $R_G(H, H')$ that lies in $H \setminus H'$. Again, since

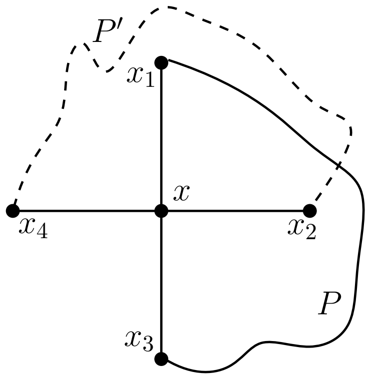


Figure 2.1: Non-piercing implies cross-free in $\mathbb{R}^2$.

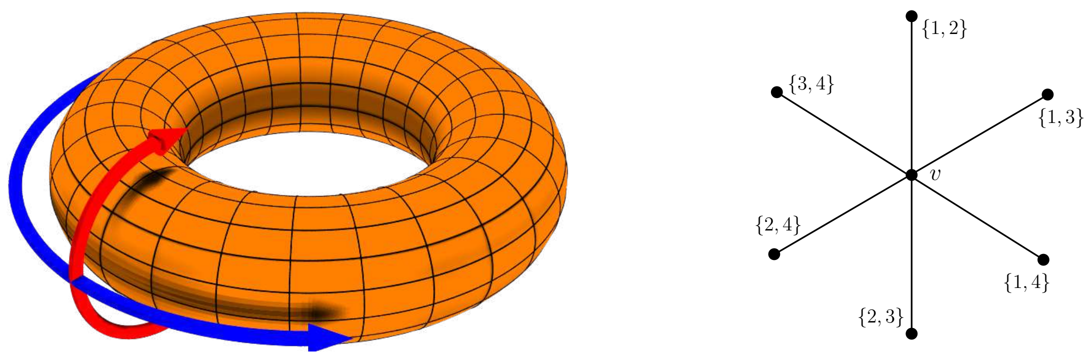
$\mathcal{H}$ is non-piercing, $H' \setminus H$ induces a connected subgraph of G. Therefore, there is a path P' between x_2 and x_4 that lies in $H' \setminus H$ (see Fig. 2.1). Observe that $P \cup \{x_1, x\} \cup \{x, x_3\}$ induces a Jordan curve with x_2 and x_4 on either side of it. Thus P and P' intersect at a vertex that lies in $H \cap H'$, which is not possible since P and P' are disjoint. Therefore, there is no path P' between x_2 and x_4 in $H' \setminus H$ which implies $H' \setminus H$ is not connected; a contradiction. ∎

Note that the reverse implication does not hold. It is easy to construct examples of graph systems in the plane that are cross-free but are piercing. Consider the graph system consisting of a graph $K_{1,4}$ embedded in the plane, with central vertex v, and leaves a, b, c, d in cyclic order. Let H and H' be two subgraphs, where H is the graph induced on $\{v, a, b\}$ and H' is the graph induced on $\{v, c, d\}$. Then, H and H' are cross-free, but neither $H \setminus H'$ nor $H' \setminus H$ is connected.

The proof of Theorem 2.1 relies on the Jordan Curve Theorem, and the corresponding statement does not hold for surfaces of higher genus. For example, let G be the torus-grid graph $T_{n,n} = C_n \square C_n$[Wei16]. The subgraphs $\mathcal{H}$ are the n non-contractible cycles perpendicular to the hole, and the n non-contractible cycles parallel to the hole. Note that the system $(T_{n,n}, \mathcal{H})$ is non-piercing but not cross-free. See Fig. 2.2a. Any pair of parallel and perpendicular cycles intersect at a unique vertex, and therefore, in the dual support, the vertices corresponding to these two cycles must be adjacent. Therefore, the dual support is $K_{n,n}$ which is not embeddable on the torus for large enough n.

For a similar construction of geometric hypergraphs, consider the following example.

Example 2.1. In the above construction, replace each cycle parallel (resp. perpendicular) to the hole with a strip parallel (resp. perpendicular) to the hole, so that each cell in the arrangement of these geometric regions has a point corresponding to the vertices of G. We get a geometric hypergraph defined by non-piercing regions on the torus such that the underlying dual hypergraph does not admit a sparse support.



(a) Torus-grid graph with subgraphs as cycles parallel to the hole and perpendicular to the hole. The dual support is K_n. [Figure source: Wikipedia.]

(b) $K_{1,\binom{4}{2}}$ and its subgraphs $H_1, \ldots, H_4$, where H_i contains the vertex v and the leaf vertices containing the label i. The dual support is a complete graph K_4.

Figure 2.2: Cross-free or non-piercing is essential.

The examples above show that the planar support construction of Raman and Ray [RR20] for the non-piercing regions can not be extended to hypergraphs defined by similar regions on non-planar surfaces. However, we show in Section 2.5 that cross-free is a sufficient condition for the construction of primal, dual and intersection supports.

▷ Why cross-free or non-piercing?.

The following sections of this chapter are devoted to the construction of bounded genus support for cross-free graph systems. We also show in Chapters 3 and 4 that non-piercing is a sufficient condition to obtain a support of bounded treewidth. The examples below show that if we drop the cross-free (respectively, non-piercing) assumption, the underlying graph system need not admit a support of bounded genus (respectively, bounded treewidth) even if the host graph is a tree.

Example 2.2 (Primal)**.** Let G be a star on $n+1$ vertices with centre v, and let $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ be a 2-coloring, where v colored is $\boldsymbol{r}$ and all other vertices are colored $\mathbf{b}$. Let $\mathcal{H}$ be a collection of all the $\binom{n}{2}$ paths on three vertices, each containing v. The graph system $(G, \mathcal{H})$ is neither cross-free nor non-piercing. The primal support is the complete graph K_n since for each pair of vertices u, w that are colored **b**, there is an $H \in \mathcal{H}$ containing exactly u, w and v, and hence requires an edge between u and w in any primal support.

Note 2.3.1. We showed in Theorem 1.1 that a hereditary class of hypergraphs admitting a sparse support has bounded VC-dimension. We also pointed out in Note 1.2.1 that being hereditary is necessary. The example above contrasts the converse argument of Note 1.2.1. In particular, each hyperedge (subgraph) consists of three vertices. The VC-dimension is 2 since the vertex v is common to all the hyperedges, but the support is a complete graph.

Example 2.3 (Dual)**.** Let $G = K_{1,\binom{n}{2}}$ be a star with center v. Each leaf vertex v_j is labeled with a unique pair $\{x_j, y_j\}$, where $\{x_j, y_j\} \subseteq \{1, 2, \ldots, n\}$. Let $\mathcal{H}$ consists of n connected subgraphs $H_1, H_2, \ldots, H_n$, where $H_i = \{v\} \cup \{v_j : i \in \{x_j, y_j\}\}$. Again, the graph system $(G, \mathcal{H})$ is neither cross-free nor non-piercing. For each pair of subgraphs in $\mathcal{H}$, there is a vertex in G that is contained in exactly those two subgraphs. This pair is adjacent in every dual support and hence, the dual support is K_n. Fig. 2.2b shows an example for $n = 4$.

We end this section with a remark to visualize the basic terminology before proceeding to the construction of supports.

Remark 2.3.1. Recall Fig. 1.5 illustrating the notions of primal, dual, intersection hypergraphs, and their respective supports. The subgraphs in $\mathcal{H}$ are non-piercing and hence cross-free as shown in Fig. 1.5b. In Fig. 1.5c, the subgraphs $H'_1, H'_2, H'_3 \in \mathcal{H}'$ are pairwise non-piercing; however, the pair of subgraphs H'_2, H'_4 is cross-free but not non-piercing.

§ 2.4 Tools to Construct Support with Bounded Genus

In this section, we explain the central machinery for the construction of support. In Section 2.4.1, we introduce the notion of *vertex bypassing*, which is crucial for the construction of a support of bounded genus. Recall from Chapter 1 that for a family $\mathcal{H}$ of subgraphs of $G = (V, E)$, we denote $\mathcal{H}_v = \{H \in \mathcal{H} : v \in H\}$ for $v \in V$, and $\mathcal{H}_e = \{H \in \mathcal{H} : e \in H\}$ for $e \in E$. We use $|.|$ to denote the cardinality of a set and define the depth of a vertex as $\text{DEPTH}(v) = |\mathcal{H}_v|$. Similarly for an edge e of G, $\text{DEPTH}(e) = |\mathcal{H}_e|$. The operation vertex bypassing modifies the subgraphs locally around a vertex v, and the resulting subgraphs may become disconnected. Then we argue in Section 2.4.2 that we can add a set of non-crossing edges so that the resulting induced subgraphs become connected while preserving the embedding of the resulting graph on the same surface. During the construction of support, we iteratively decrease the depth of a vertex using the vertex bypassing operation and then use induction to complete our argument.

§ 2.4.1 Vertex bypassing

Vertex Bypassing (VB(v)) takes a cross-free system $(G, \mathcal{H})$ as input and *simplifies* the system around a vertex v of G. For an embedded cross-free graph system $(G, \mathcal{H})$, Vertex bypassing at a vertex v is defined as follows:

Definition 2.6 (VB(v)). Let $(G, \mathcal{H})$ be a cross-free system, and that we have a cross-free embedding of G with respect to $\mathcal{H}$ in an oriented surface Σ. Let $N(v) = (v_0, \ldots, v_{k-1}, v_0)$ be the cyclic order of neighbors of v in that embedding.

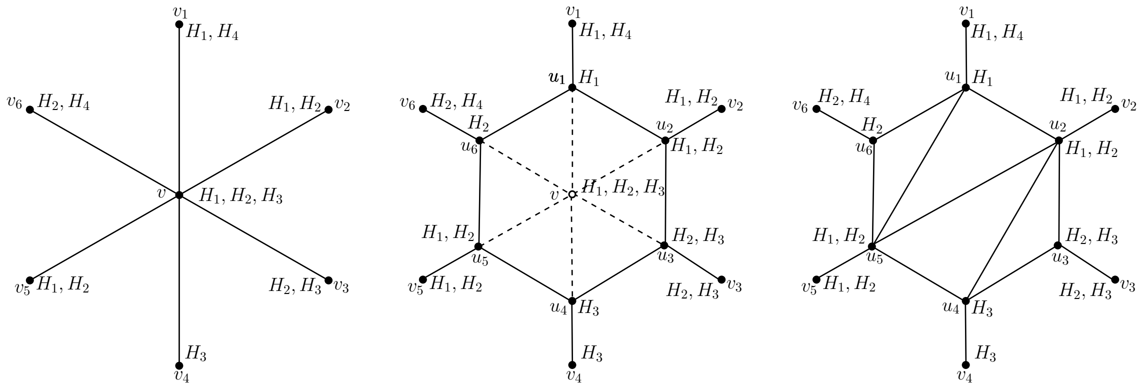
Step 1: Subdivide each edge $\{v, v_i\}$ by a vertex u_i. Remove the vertex v and connect consecutive vertices u_i, u_{i+1} (with indices taken $\bmod k$) with a simple arc not intersecting the edges of G to construct a cycle $C = (u_0, \ldots, u_{k-1}, u_0)$ so that the resulting graph G'' remains embedded on Σ.

Step 2: For $H \in \mathcal{H}_v$, let H' denote the subgraph of G'' induced by $(V(H) \setminus \{v\}) \cup \{u_i : v_i \in V(H)\}$. Let $\mathcal{H}'_v = \{H' : H \in \mathcal{H}_v\}$, and finally $\mathcal{H}' = (\mathcal{H} \setminus \mathcal{H}_v) \cup \mathcal{H}'_v$ (Note that the subgraphs in $\mathcal{H}'_v$ may not be connected).

Step 3: Add a set D of internally non-intersecting chords in C so that $\forall H \in \mathcal{H}'$, H induces a connected subgraph in $C \cup D$, and the resulting graph system $(G', \mathcal{H}')$ remains cross-free.

Fig. 2.3 shows vertex bypassing at vertex v. It is easy to see that the graph G' obtained from G is also embedded in Σ as each step preserves the embedding. Since we removed vertex v in Step 1, the subgraphs $\mathcal{H}'_v$ of G'' may become disconnected. The main challenge is to add additional edges to the graph G'' so that the resulting subgraphs $\mathcal{H}'$ of G' become connected, and the graph system $(G', \mathcal{H}')$ remains cross-free.

To achieve this, we introduce the notion of $abab$-free hypergraphs. An equivalent notion, namely $ABAB$-free hypergraphs, was studied by Ackerman et al. [AKP20], where the elements of the hypergraph are placed in a linear order instead of a cyclic order (though the cyclic and linear ordering are equivalent for $ABAB$-freeness).



(a) Vertices in $N(v)$. **(b)** Cycle C on u_i's. **(c)** Chords in C.

Figure 2.3: Vertex Bypassing

Definition 2.7 ($abab$-free hypergraph)**.** An abstract hypergraph $(X, \mathcal{H})$ is said to be $abab$-free if the elements of X can be arranged on a cycle C such that for any hyperedges $H, H' \in \mathcal{H}$, there does not exist distinct $x_1, x_2, x_3, x_4 \in X$ in this order around C such that $x_1, x_3 \in H \setminus H'$, and $x_2, x_4 \in H' \setminus H$.

We will revisit the abstract $abab$-free hypergraphs in Chapter 7. In the present chapter however, since C is a graph that is a cycle, we use the term $abab$-free subgraph to mean the hyperedges of $\mathcal{H}$. The relation between cross-free graph systems and $abab$-free systems is the following: If G is embedded so that it is cross-free with respect to $\mathcal{H}$, then for any vertex v, on applying VB(v), the system $(C, \mathcal{H}'_v)$ is $abab$-free, where C is the cycle on the vertices $u_0, \ldots, u_{deg(v)-1}$ subdividing the edges $\{v, v_i\}$.

Proposition 2.1. Let $(G, \mathcal{H})$ be a cross-free graph system with a cross-free embedding with respect to $\mathcal{H}$. For any vertex $v \in V(G)$, on applying VB(v), $(C, \mathcal{H}'_v)$ is $abab$-free.

Proof. Since the subgraphs in $\mathcal{H}$ are induced subgraphs, $\mathcal{H}_e \subseteq \mathcal{H}_v$. Consider the cycle $C = (u_0, \ldots, u_{deg(v)-1})$ on the subdivided vertices added on applying VB(v). By construction, $\mathcal{H}_{u_i} = \mathcal{H}_e$, where $e = \{v, v_i\}$, and $\mathcal{H}_e \subseteq \mathcal{H}_v$. If $(C, \mathcal{H}'_v)$ is not $abab$-free, let H_1, H_2 be subgraphs in $\mathcal{H}'_v$ that are not $abab$-free. This implies that H_1 and H_2 are crossing at v. ∎

By Proposition 2.1 therefore, the problem of adding a set of non-intersecting chords D in Step 3 of vertex-bypassing reduces to the following question.

Question 2.1. Given an $abab$-free embedding of an $abab$-free hypergraph $\mathcal{H}$ on a cycle C. Can we add a set D of non-intersecting chords in C such that each hyperedge induces a connected subgraph of $C \cup D$?

We give a positive answer to this question in the following lemma, whose proof is in Section 2.4.2.

Lemma 2.1. Let C be a cycle embedded in the plane, and let $\mathcal{K}$ be a set of $abab$-free subgraphs of C. Then, we can add a set D of non-intersecting chords in C such that each $K \in \mathcal{K}$ induces a connected subgraph of $C \cup D$. Further, the set D of non-intersecting chords to add can be computed in time $O(mn^4)$, where $m = |\mathcal{K}|$, and $n = |C|$.

With Lemma 2.1 in hand, we can obtain the desired cross-free system $(G', \mathcal{H}')$.

Lemma 2.2. Let $(G, \mathcal{H})$ be a cross-free system with a cross-free embedding of G with respect to $\mathcal{H}$. Suppose we apply $\mathrm{VB}(v)$ to a vertex $v \in V(G)$. Then, in the resulting graph system $(G', \mathcal{H}')$, each $H \in \mathcal{H}'$ is connected. Further, $\mathrm{VB}(v)$ can be done in time $O(mn^4)$, where $m = |\mathcal{H}|$ and $n = |V(G)|$.

Proof. Let C be the cycle added on the subdividing vertices around vertex v. Since $(G, \mathcal{H})$ is cross-free, by Proposition 2.1, the subgraphs $\{H \cap C : H \in \mathcal{H}'_v\}$ satisfy the $abab$-free property on C. Therefore, by Lemma 2.1, there is a collection D of non-intersecting chords such that each subgraph in $\mathcal{H}'_v$ induces a connected subgraph of $C \cup D$. Hence, each subgraph $H \in \mathcal{H}'$ is a connected subgraph of G' since each $H \in \mathcal{H}_v$ is modified only in the vertices of subdivision. Since Lemma 2.1 guarantees that the set D of non-intersecting chords to add can be computed in time $O(|\mathcal{H}||V(G)|^4)$, it follows that $\mathrm{VB}(v)$ can be done in time $O(mn^4)$. ■

In the following, we argue that if $(G, \mathcal{H})$ is cross-free, then the resulting system $(G', \mathcal{H}')$ obtained on applying $\mathrm{VB}(v)$ at a vertex $v \in V(G)$ is cross-free.

Lemma 2.3. Let $(G, \mathcal{H})$ be a cross-free system. Let $(G', \mathcal{H}')$ be the system obtained after applying vertex bypassing at a vertex $v \in V(G)$. Then, $(G', \mathcal{H}')$ is cross-free.

Proof. By Lemma 2.2, the subgraphs in $\mathcal{H}'$ are connected subgraphs of G'. We argue that $(G', \mathcal{H}')$ is cross-free. Let the cyclic sequence of the neighbors of v be $(v_0, \ldots, v_{k-1})$. Similarly, let $(u_0, \ldots, u_{k-1})$ denote the cyclic sequence of the vertices in G', where u_i is the vertex subdividing the edge $\{v, v_i\}$ in $\mathrm{VB}(v)$.

Consider two subgraphs $H_1, H_2 \in \mathcal{H}$. We will show that H'_1 and H'_2 are cross-free, where H'_1 and H'_2 are respectively, the subgraphs in $\mathcal{H}'$ corresponding to H_1 and H_2

in $\mathcal{H}$. For a vertex $x \in V(G)$, let x' denote its copy in $V(G')$. We let $\tilde{x}$ denote its corresponding vertex in the reduced graph $R_G(H_1, H_2)$, and $\tilde{x'}$, the vertex in $R_{G'}(H_1', H_2')$ corresponding to vertex $x' \in V(G')$.

For $\tilde{x} \in V(R_G(H_1, H_2))$, let $(A_0, \ldots, A_\ell - 1)$ be the *cyclic-pattern* around $\tilde{x}$, where each A_i is a subset of $\{H_1, H_2\}$, i.e., if $\tilde{z}_0, \ldots, \tilde{z}_{\ell-1}$ are the neighbors of $\tilde{x}$ in cyclic order, then A_i corresponds to the subset of $\{H_1, H_2\}$ containing $\tilde{z}_i$. We use an identical notation for $R_{G'}(H_1', H_2')$.

Consider an $x' \in V(G')$ and its corresponding vertex in $R_{G'}(H_1', H_2')$. First, suppose $v \notin V(H_1) \cap V(H_2)$. If $x' \notin \{u_0, \ldots, u_{k-1}\}$, then the cyclic-pattern at $\tilde{x'}$ in $R_{G'}(H_1', H_2')$ is isomorphic to the cyclic-pattern at $\tilde{x}$ (replacing H_j' by H_j, $j = 1, 2$), where x is the vertex in G corresponding to x'. Since H_1 and H_2 are cross-free at x, so are H_1' and H_2' at x'. Since the subgraphs at u_i are a subset of the subgraphs at v, it implies that no u_i is contained in both H_1 and H_2. Hence, H_1 and H_2 are cross-free at each u_i.

Now, if $v \in V(H_1) \cap V(H_2)$, then for any vertex y' in G' so that its copy y in G is not connected to v via a path that lies in $V(H_1) \cap V(H_2)$, the cyclic-pattern at $\tilde{y'}$ and $\tilde{y}$ are isomorphic. If y lies on a path in $V(H_1) \cap V(H_2)$ to v, then $\tilde{y} = \tilde{v}$ in $R_G(H_1, H_2)$. In this case, observe that the cyclic pattern at $\tilde{y'}$ is a sub-sequence of the cyclic pattern at $\tilde{y}$ since $\tilde{y'} = \tilde{u}_i$ for some $u_i \in H_1' \cap H_2'$. Since H_1 and H_2 are non-crossing at v, it implies that H_1' and H_2' are non-crossing at y'.

Since H_1' and H_2' are non-crossing at all vertices of G', and $H_1, H_2 \in \mathcal{H}$ were arbitrary, it implies $(G', \mathcal{H}')$ is cross-free. ∎

§ 2.4.2 Non-blocking chords in $abab$-free hypergraphs

In this section, we prove Lemma 2.1. Let $C = (x_0, \ldots, x_{k-1}, x_0)$ be a cycle embedded in the plane with vertices labeled in clockwise order. Let $\mathcal{K}$ be a collection of subgraphs of C such that $\mathcal{K}$ is *abab*-free. For $i, j \in \{0, \ldots, k-1\}$, let $[x_i, x_j]$ denote the vertices on the arc from i to j in clockwise order with indices taken $\bmod k$. Similarly, we use (x_i, x_j) to denote the open arc, i.e., consisting of the vertices on the arc from i to j except x_i and x_j. The half-open arc $(x_i, x_j]$ that excludes x_i but includes x_j, is defined similarly.

The addition of a chord $d = \{x_i, x_j\}$ divides C into two open arcs - (x_i, x_j) and (x_j, x_i). The chord d *blocks* a subgraph $K \in \mathcal{K}$ if both open arcs contain a run of K, and neither end-point of d is contained in K. Here, a *run* refers to a connected component of the subgraph K in C, i.e., a maximal sequence of consecutive vertices of K on C. Such a chord d is called a *blocking chord*. If d does not block any subgraph in $\mathcal{K}$, it is called a *non-blocking* chord. We show in Lemma 2.4 that there always exists a non-blocking chord d that connects two disjoint runs of some subgraph $K \in \mathcal{K}$.

Lemma 2.4. Let C be a cycle embedded in the plane, and let $\mathcal{K}$ be a collection of $abab$-free subgraphs in the embedding of C. Then, for some disconnected $K \in \mathcal{K}$, there exists a non-blocking chord joining two disjoint runs of K. Further, such a chord can be computed in time $O(mn^3)$, where $m = |\mathcal{K}|$ and $n = |C|$.

Proof. Assume without loss of generality that each subgraph $K \in \mathcal{K}$ induces at least two runs in C, and no two subgraphs contain the same subset of vertices of C. Define a partial order $\prec_C$ on $\mathcal{K}$, where for $K, K' \in \mathcal{K}$, $K \prec_C K'$ iff $K \cap C \subsetneq K' \cap C$. Let $K_0 \in \mathcal{K}$ be a minimal subgraph with respect to the order $\prec_C$.

Let $K_0^0, \ldots, K_0^q$ denote the runs of K_0. We let A denote the run K_0^0 and let $B = \cup_{i=1}^q K_0^i$. For ease of exposition, we assume C is drawn such that A lies in the lower semi-circle of C and that B lies in the upper semi-circle of C, where the runs $K_0^1, \ldots, K_0^q$ appear in counter-clockwise order. Let $a_0, \ldots, a_r$ denote the vertices of A in clockwise order and let $b_0, \ldots, b_s$ denote the vertices of B in counter-clockwise order. See Fig. 2.4a.

We show that there is a chord d from a vertex in A to a vertex in B that is non-blocking. In order to do so, we start with the chord $d_0 = a_0 b_0$, and construct a sequence of chords until we either find a non-blocking chord, or we end up with the chord $d_k = a_r b_s$, which will turn out to be non-blocking. Having constructed chords $d_0, \ldots, d_{i-1}$, where $d_{i-1} = a_\ell b_j$, d_i will be either the chord $a_\ell b_{j'}$ or $a_{\ell'} b_j$, where $j' > j$ and $\ell' > \ell$.

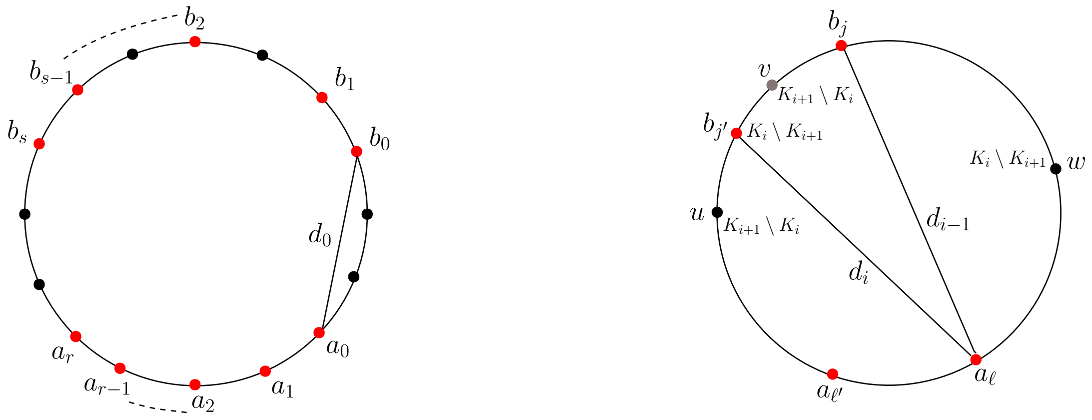
Next, we describe the construction of the chords. Each chord d we construct satisfies the following invariant: If K is a subgraph blocked by a chord $d = a_\ell b_j$, then

(i) The vertices of K are contained in the vertices of K_0 in the arc (a_ℓ, b_j), and

(ii) There is a vertex $k \in K \setminus K_0$ in the arc (b_j, a_ℓ).

Let d_0 denote the chord $a_0 b_0$. If d_0 is non-blocking, we are done. Otherwise, if any $K_1 \in \mathcal{K}$ is blocked by d_0, there is a vertex $k \in K_1$ that lies in (b_0, a_0). Since (b_0, a_0) does not contain a vertex of K_0, this implies $k \in K_1 \setminus K_0$, and hence d_0 satisfies condition (ii) of the invariant. Since we assumed the subgraphs $\mathcal{K}$ are $abab$-free, this implies that any vertex of K_1 in arc (a_0, b_0) is contained in K_0. This ensures that condition (i) of the invariant is satisfied by d_0.

Having constructed $d_0 = a_0 b_0, \ldots, d_{i-1} = a_\ell b_j$, each of which satisfy conditions (i) and (ii) of the invariant, we construct d_i as follows: We simultaneously scan the vertices of B in counter-clockwise order from b_j, and the vertices of A in clockwise order from a_ℓ until we find the first vertex x that belongs to a subgraph blocked by d_{i-1}. Let K_i denote this subgraph. If $x = b_{j'} \in B$, we set $d_i = a_\ell b_{j'}$. Otherwise, $x = a_{\ell'} \in A$, and we set $d_i = a_{\ell'} b_j$. Assume without loss of generality that $d_i = a_\ell b_{j'}$ (the other case is similar).



(a) Ordering the vertices of K_0 in sets A and B. Here, $A = \{a_0, \ldots, a_r\}$ and $B = \{b_0, \ldots, b_s\}$.

(b) Adding chords between A and B. If $K_{i+1} \setminus K_0 \neq \emptyset$ in $(a_\ell, b_{j'})$, vertices $u, b_{j'}, v, w$ form $abab$ in K_{i+1} and K_i.

Figure 2.4: Finding a non-blocking chord to join two disjoint runs of K_0.

If d_i is a non-blocking chord, we are done. Otherwise, let K_{i+1} denote a subgraph blocked by d_i. Then, both the arcs $(a_\ell, b_{j'})$ and $(b_{j'}, a_\ell)$ contain a run of K_{i+1}, and $a_\ell, b_{j'} \in K_0 \setminus K_{i+1}$. We now show that d_i satisfies the invariant. Most of the work will go into showing that d_i satisfies condition (i) of the invariant. We show this by contradiction - If d_i does not satisfy invariant (i), we will exhibit a pair of subgraphs violating the $abab$-free property.

Suppose d_i does not satisfy condition (i) of the invariant, that is, there is a vertex $u \in K_{i+1} \setminus K_0$ that lies in $(a_\ell, b_{j'})$. Since d_{i-1} satisfies both the conditions of the invariant, the subgraph K_i blocked by d_{i-1} is contained in K_0 in (a_ℓ, b_j). Since $(a_\ell, b_{j'}) \subsetneq (a_\ell, b_j)$, it implies $u \notin K_i$, and thus $u \in K_{i+1} \setminus K_i$. By construction of d_i, the vertex $b_{j'} \in K_i$, and since d_i blocks K_{i+1}, $b_{j'} \notin K_{i+1}$. Thus, $b_{j'} \in K_i \setminus K_{i+1}$.

Now, we claim that K_{i+1} is not blocked by d_{i-1}. To see this, since d_{i-1} satisfies condition (i) of the invariant, for any subgraph K' blocked by d_{i-1}, we have that $K' \subseteq K_0$ in (a_ℓ, b_j). Since $(a_\ell, b_{j'}) \subsetneq (a_\ell, b_j)$, combined with the facts that $u \in K_{i+1} \setminus K_0$ and that u lies in $(a_\ell, b_{j'})$ implies that K_{i+1} is not blocked by d_{i-1}. But K_{i+1} is blocked by d_i; it follows that there is a vertex v of K_{i+1} in the arc $(b_{j'}, b_j]$. Note that v need not lie in K_0. However, no vertex in the arc $(b_{j'}, b_j]$ lies in K_i, since $b_{j'}$ was the first vertex encountered that was contained in a subgraph blocked by d_{i-1} when traversing the vertices of B in counter-clockwise order from b_j. Therefore, $v \in K_{i+1} \setminus K_i$.

Finally, since d_{i-1} satisfies condition (ii) of the invariant, there is a vertex $w \in K_i \setminus K_0$ that lies in (b_j, a_ℓ). Note that $u \in K_{i+1} \setminus K_0$, $a_\ell, b_{j'} \in K_0 \setminus K_{i+1}$, and u lies in $(a_\ell, b_{j'})$. There is no vertex $x \in K_{i+1} \setminus K_0$ that lies in $(b_{j'}, a_\ell)$, since otherwise we have an $abab$-pattern among K_{i+1} and K_0 given by the cyclic sequence of the vertices $u, b_{j'}, x, a_\ell$.

So, we have $K_{i+1} \subseteq K_0$ in $(b_{j'}, a_\ell)$ since the arrangement is $abab$-free. However, since $(b_j, a_\ell) \subsetneq (b_{j'}, a_\ell)$ and $w \in K_i \setminus K_0$ lies in (b_j, a_ℓ), it follows that $w \notin K_{i+1}$. Therefore, $w \in K_i \setminus K_{i+1}$. See Fig. 2.4b.

From the above arguments, it follows that the subgraphs K_{i+1} and K_i are not $abab$-free, as witnessed by the sequence of vertices $u, b_{j'}, v$ and w, a contradiction. Thus, d_i satisfies condition (i) of the invariant. The fact that d_i satisfies condition (ii) of the invariant follows from the fact that K_0 is minimal. Otherwise, $K_{i+1} \subseteq K_0$ in $(a_\ell, b_{j'})$ and in $(b_{j'}, a_\ell)$, and therefore $K_{i+1} \subsetneq K_0$.

Since the set of chords is finite, the sequence of chords constructed either ends in a non-blocking chord, or we end up with the chord $d = a_r b_s$. We claim that d must be a non-blocking chord. Suppose d blocks a subgraph K. Then, (a_r, b_s) contains a vertex $u \in K \cap K_0$, as d satisfies invariant (i) and (ii). However, (a_r, b_s) does not contain a vertex in K_0. Therefore, d must be a non-blocking chord.

For any chord d_0 joining two disjoint runs of some K_0, it can be tested in $O(mn)$ if d_0 is a non-blocking chord since any $K \in \mathcal{K}$ has at most n vertices. We scan over at most $\binom{n}{2}$ chords, and the lemma ensures one of these chords must be non-blocking. Hence, a non-blocking chord can be computed in $O(mn^3)$ time. ■

We are now ready to prove Lemma 2.1. We do this by using Lemma 2.4 to add a non-blocking chord connecting two disconnected components of a subgraph, and then recursively apply Lemma 2.4 to the two resulting cycles and their induced subgraphs.

Lemma 2.1. Let C be a cycle embedded in the plane, and let $\mathcal{K}$ be a set of $abab$-free subgraphs of C. Then, we can add a set D of non-intersecting chords in C such that each $K \in \mathcal{K}$ induces a connected subgraph of $C \cup D$. Further, the set D of non-intersecting chords to add can be computed in time $O(mn^4)$, where $m = |\mathcal{K}|$, and $n = |C|$.

Proof. For a subgraph $K \in \mathcal{K}$, let n_K denote the number of disjoint runs of K on C. Let

$$\mathrm{cost}(C, \mathcal{K}) = \sum_{K \in \mathcal{K}} (n_K - 1)$$

If $\mathrm{cost}(C, \mathcal{K}) = 0$, then every subgraph $K \in \mathcal{K}$ consists of one run, and therefore $C \cap K$ is connected for each $K \in \mathcal{K}$, and $D = \emptyset$ suffices.

Suppose the lemma holds for all $(C', \mathcal{K}')$ with $\mathrm{cost}(C', \mathcal{K}') < N$. Now, consider an instance with an embedded cycle C and a set $\mathcal{K}$ of $abab$-free subgraphs of C such that $\mathrm{cost}(C, \mathcal{K}) = N$. By Lemma 2.4, there is a non-blocking chord $d = \{x, y\}$ joining two disjoint runs of some subgraph $K_0 \in \mathcal{K}$.

The chord $d = \{x, y\}$ divides the cycle C into two arcs, $[x, y]$, and $[y, x]$. We construct two disjoint sub-problems on the cycles C_ℓ and C_r obtained from C, where C_ℓ is obtained by adding the edge $\{x, y\}$ to the arc $[y, x]$, and C_r is obtained by adding the edge $\{x, y\}$ to the arc $[x, y]$. The subgraphs in C_ℓ and C_r are respectively those induced by the subgraphs in $\mathcal{K}$, namely $\mathcal{K}_\ell = \{K \cap C_\ell : K \in \mathcal{K}\}$, and $\mathcal{K}_r = \{K \cap C_r : K \in \mathcal{K}\}$. Note that $\mathcal{K}_\ell$ and $\mathcal{K}_r$ are $abab$-free on C_ℓ and C_r, respectively. Let n_K^ℓ and n_K^r denote respectively, the number of runs of a subgraph $K \in \mathcal{K}$ in C_ℓ and in C_r. Clearly, $n_{K_0}^\ell < n_{K_0}$ and $n_{K_0}^r < n_{K_0}$ as d joins two disconnected runs of K_0. Also, for all other subgraphs $K' \in \mathcal{K}$, $n_{K'}^\ell \leq n_{K'}$ and $n_{K'}^r \leq n_{K'}$, it follows that $\mathrm{cost}(C_r, \mathcal{K}_r) < \mathrm{cost}(C, \mathcal{K})$ and $\mathrm{cost}(C_\ell, \mathcal{K}_\ell) < \mathrm{cost}(C, \mathcal{K})$.

Hence, by the inductive hypothesis on the pair $(C_\ell, \mathcal{K}_\ell)$, there exists a set of non-intersecting chords D_ℓ such that each $K \in \mathcal{K}_\ell$ induces a connected subgraph of $C_\ell \cup D_\ell$. Similarly, there exists a set of non-intersecting chords D_r such that each $K \in \mathcal{K}_r$, induces a connected subgraph of $C_r \cup D_r$. Let $D = D_\ell \cup D_r \cup d$. If there is a subgraph $K \in \mathcal{K}$ such that K induces connected subgraphs on both $C_r \cup D_r$ and $C_\ell \cup D_\ell$ but K is not connected on $C \cup D$, then $K \cap \{x, y\} = \emptyset$. It follows that K is blocked by d; a contradiction that d is a non-blocking chord. Hence, it D is a set of non-intersecting chords such that each $K \in \mathcal{K}$ induces a connected subgraph of $(C \cup D)$.

By Lemma 2.4, a non-blocking chord can be computed in $O(mn^3)$ time. Since $C \cup D$ gives an outerplanar graph, $|D| \leq n - 3$. Hence, the total running time to compute D is $O(mn^4)$. ∎

§ 2.5 Construction of Bounded Genus Supports

In this section, we show that for cross-free systems on a graph of genus g, there exist primal, dual and intersection supports of genus at most g. While the existence of an intersection support implies the existence of the primal and dual supports, we use the dual support and techniques from the primal support in order to construct the intersection support, and hence present these first.

Before we proceed further, we prove a lemma below, which we will be using in Chapters 3 and 4 as well, for the construction of dual and intersection supports. We emphasize that in the lemma, G is any arbitrary graph, and $\mathcal{H}, \mathcal{K}$ are any sets of subgraphs of G. Let $\mathcal{G}$ denote a graph class that is closed under adding vertices of degree one, i.e., if $G_1 \in \mathcal{G}$, and G_2 is obtained from G_1 by adding a new vertex and making it adjacent to one of the vertices of G_1, then $G_2 \in \mathcal{G}$.

Lemma 2.5. Let $(G, \mathcal{H}, \mathcal{K})$ be an intersection system, and $H, H' \in \mathcal{H}$ be two subgraphs such that $H \subsetneq H'$. Set $\mathcal{H}' = \mathcal{H} \setminus \{H\}$. If $(G, \mathcal{H}', \mathcal{K})$ admits an intersection support $Q' \in \mathcal{G}$, then $(G, \mathcal{H}, \mathcal{K})$ also admits an intersection support $Q \in \mathcal{G}$.

Proof. Let Q' be any fixed intersection support for $(G, \mathcal{H}', \mathcal{K})$. Let $Q = Q' + e$, where e is the edge in Q joining H and H'. In other words, Q is obtained from Q' by adding a new vertex H, and making it adjacent to H'. Clearly, $Q \in \mathcal{G}$. Let $K \in \mathcal{K}$ be arbitrary. We need to show that $\mathcal{H}_K$ induces a connected subgraph of Q. If $K \cap H = \emptyset$, then $\mathcal{H}_K = \mathcal{H}'_K$, which induces a connected subgraph of Q', and hence of Q. So, assume that $K \cap H \neq \emptyset$. Then, $K \cap H' \neq \emptyset$ since $H \subsetneq H'$. In this case, $\mathcal{H}_K = \mathcal{H}'_K \cup \{H\}$. Now the fact that $\mathcal{H}_K$ induces a connected subgraph of Q follows from the fact that $\mathcal{H}'_K$ induces a connected subgraph of Q' and that H is adjacent to H' in Q. ∎

By Lemma 2.5 above, if $(G, \mathcal{H}, \mathcal{K})$ is any graph system, and the goal is to construct an intersection support $Q \in \mathcal{G}$, then it is sufficient to construct a support when the elements of $\mathcal{H}$ are pairwise incomparable. If elements of $\mathcal{H}$ are pairwise incomparable, we say that $\mathcal{H}$ is *containment-free*. The lemma also applies to the dual hypergraphs since they are a special case of intersection hypergraphs. Since adding vertices of degree one to a graph does not increase the genus of the resulting graph, Lemma 2.5 applies to graph systems of genus g. For completeness, we state a corollary below for the dual hypergraph arising from a cross-free graph system of genus g. The proof immediately follows from Lemma 2.5.

Corollary 2.2. Let $(G, \mathcal{H}, \mathcal{K})$ be an intersection system. To construct a dual support for $(G, \mathcal{H})$ or an intersection support for $(G, \mathcal{H}, \mathcal{K}))$ that has genus at most g, it is sufficient to construct such a support when $\mathcal{H}$ is containment-free.

The classes of outerplanar graphs and graphs of treewidth t are also closed under adding degree-one vertices. We will make use of the lemma in Chapters 3 and 4 when dealing with these graph classes.

From now onwards in this chapter, G is a graph of genus g. If a subgraph $H \in \mathcal{H}$ contains an edge e, then it contains both the end-points of e. Hence, $\mathcal{H}_e \subseteq \mathcal{H}_v$ for any $v \in V$ and $e \sim v$, where $e \sim v$ denotes that the edge e is incident to v. We say that a vertex $v \in V(G)$ is *maximal* if $\text{DEPTH}(v) > \text{DEPTH}(e)$ for all $e \sim v$. We need the following basic result that follows from vertex bypassing and will be used for inductive arguments for the construction of a dual support.

Lemma 2.6. Let $(G, \mathcal{H})$ be a cross-free graph system and let $v \in V(G)$ be maximal. Let $(G', \mathcal{H}')$ be the graph system obtained on applying $\text{VB}(v)$. Then, $\text{DEPTH}(u_i) < \text{DEPTH}(v)$ and u_i is not a maximal vertex for all u_i sub-dividing the edge $\{v, v_i\}$ for $v_i \in N(v)$.

Proof. Let $e_i = \{v, v_i\}$ be the edge subdivided by the vertex u_i. Since v is maximal, $\mathcal{H}_{e_i} \subsetneq \mathcal{H}_v$, i.e., $\text{DEPTH}(e_i) < \text{DEPTH}(v)$. Also, note that by construction, $\text{DEPTH}(u_i) = \text{DEPTH}(e_i)$. It follows that $\text{DEPTH}(u_i) < \text{DEPTH}(v)$ and that u_i is not a maximal vertex. Since u_i was chosen arbitrarily, we have $\text{DEPTH}(u_i) < \text{DEPTH}(v)$ for all $i = 1, 2, \ldots, deg(v)$. ∎

§ 2.5.1 Primal support

In this section, we show that for a cross-free system $(G, \mathcal{H})$ of genus g with $c : V \to \{\mathbf{b}, \boldsymbol{r}\}$, there is a primal support Q of genus at most g. We start with the following lemma by showing that if no red vertex is maximal and the subgraphs in $\mathcal{H}$ are connected (and not necessarily cross-free), then it is easy to construct a desired primal support. Our construction of a support in the general case follows by repeated application of the Vertex Bypassing operation to a maximal red vertex. We show that each Vertex Bypassing operation decreases the number of maximal red vertices, and hence we eventually obtain a system with no maximal red vertices. While the lemma below works for any hereditary family of graphs closed under edge-contractions, we state it only for the graphs of bounded genus, as this is the statement required for subsequent theorems.

A pair of vertices $u, v \in V(G)$ are said to be *twins* if $\mathcal{H}_u = \mathcal{H}_v$. We say that u and v are *adjacent twins* if in addition u and v are adjacent vertices in G. If $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ is a 2-coloring of $V(G)$ and $u, v \in \boldsymbol{r}(V(G))$ are twins, we say that u and v are *red twins*.

Lemma 2.7. Let $(G, \mathcal{H})$ be a graph system of genus g such that each $H \in \mathcal{H}$ is a connected subgraph of G. Let $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ be any 2-coloring of $V(G)$ such that no vertex in $\boldsymbol{r}(V(G))$ is maximal and there are no adjacent twins $u, v \in \boldsymbol{r}(V(G))$. Then there is a primal support of genus at most g.

Proof. We prove by induction on $|\boldsymbol{r}(V(G))|$. If $|\boldsymbol{r}(V(G))| = 0$, then $Q = G$ is a desired support. Suppose the statement holds for any graph system satisfying the conditions in the lemma with fewer than k red vertices.

Let $(G, \mathcal{H})$ be a graph system with a 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ satisfying the conditions of the lemma such that $|\boldsymbol{r}(V(G))| = k$.

For each vertex $u \in \boldsymbol{r}(V(G))$, we do the following: Since u is not maximal, by definition, there is an edge $e = \{u, v\} \in E(G)$ such that $\mathcal{H}_e = \mathcal{H}_u$. We chose one such edge e incident to u, breaking ties arbitrarily. Since $\mathcal{H}_e \subseteq \mathcal{H}_v$, it follows that $\mathcal{H}_u \subseteq \mathcal{H}_v$. We orient this edge from u to v. Further, if $v \in \boldsymbol{r}(V(G))$, then $\mathcal{H}_u \subsetneq \mathcal{H}_v$ since there are no adjacent twins in $\boldsymbol{r}(V(G))$.

It follows that the set of oriented edges induces a directed acyclic subgraph. Further, since there are no twins in $\boldsymbol{r}(V(G))$ and no red vertex has out-degree more than one, this directed acyclic graph is an oriented forest. Since no red vertex is maximal, it implies that each tree in the forest has its edges oriented towards the root, and the root must be a blue vertex. Choose an arbitrary tree in the forest and choose an edge $e = \{u, v\}$ within the tree, where v is the root of the tree. Let $G' = G/e$, i.e., the graph obtained by contracting the edge e with the new vertex taking the color $c(v) = \mathbf{b}$, since v, the root of the tree, is colored $\mathbf{b}$. See Fig. 2.5. Let $\mathcal{H}'$ be the resulting family of subgraphs, where for each $H \in \mathcal{H}$, we have a subgraph $H' \in \mathcal{H}'$ of G' induced on the vertices $V(H) \cap V(G')$.

Note that no red vertex is maximal in G' since each red vertex in G' has an outgoing edge incident on it. Further, there are no adjacent twins in G' among the red vertices, as the edge contraction did not introduce new adjacencies between the red vertices. To see this, note that the vertex obtained by contracting the edge $\{u, v\}$ is colored $\mathbf{b}$, and hence, the adjacency among the red vertices is unchanged. Since there are no adjacent red twins in G, there are no adjacent twins in G'. Finally, note that G' has genus at most g since contracting an edge in G preserves the embedding of the resulting graph on the same surface as that of G.

Since G' has $k-1$ red vertices, by the inductive hypothesis, there is a primal support Q on the blue vertices of G' such that Q has genus at most g. The graph Q is also a support for $(G, \mathcal{H})$ since the set of blue vertices in H remains unchanged in $(G', \mathcal{H}')$ for each $H \in \mathcal{H}$. ∎

Now we prove that given a cross-free graph system $(G, \mathcal{H})$ of genus g and a 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$, there is a primal support of genus at most g. To establish the result, we repeatedly apply edge contractions for adjacent red twins, and use an induction argument on the maximum depth of a maximal red vertex via the applications of the vertex bypassing operation in combination with Lemma 2.6.

Theorem 2.3. Let $(G, \mathcal{H})$ be a cross-free graph system of genus g. For any coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$, there exists a primal support Q of genus at most g on $\mathbf{b}(V)$.

Proof. If there are adjacent red twins u and v, we contract the edge $\{u, v\}$ to a single vertex and assign it color $\boldsymbol{r}$. This preserves the cross-freeness among subgraphs in

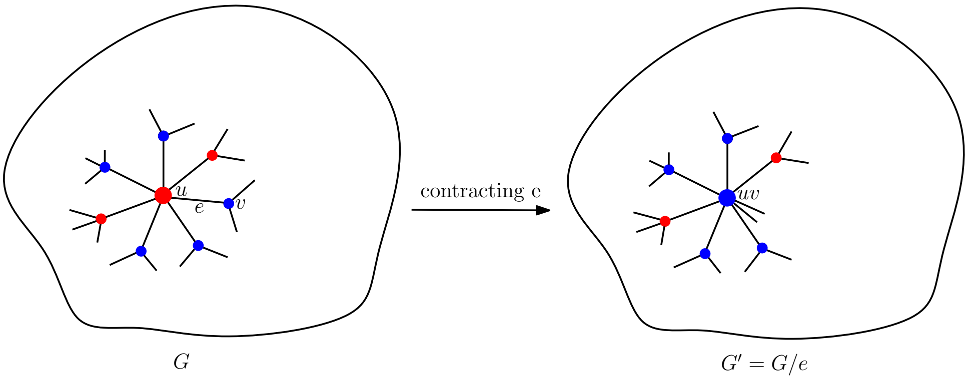


Figure 2.5: Reducing the number of red vertices by contracting an edge incident to a non-maximal red vertex u. After contracting $e = \{u, v\}$, the common vertex gets the color of v.

$\mathcal{H}$. Since for any $H \in \mathcal{H}$, the set of vertices in $\mathbf{b}(V(G)) \cap V(H)$ remains unchanged, a primal support for the modified system is also a support for the given system $(G, \mathcal{H})$. So we assume, without loss of generality, that there is no adjacent red twin.

We prove the result by induction on the maximum depth of a maximal red vertex. Let d be the maximum depth of a maximal red vertex. Note that a non-maximal red vertex can have depth more than d. For $i \in \{1, 2, \ldots, d\}$, let $S_i \subseteq \boldsymbol{r}(V(G))$ be the set of maximal red vertices of depth i. Let $d = 1$. It follows from the definition of a maximal vertex that for each $v \in S_d$ the subgraph containing v consists of a single vertex v. We remove all such subgraphs from $\mathcal{H}$, since they do not contain any blue vertex and thus the conditions of support are trivially satisfied for them. Let $\mathcal{H}_1 = \mathcal{H} \setminus S_1$. Then, the graph system $(G, \mathcal{H}_1)$ satisfies the conditions of Lemma 2.7, and thus admits a primal support.

Assume the statement holds for any cross-free graph system of genus g with $d < k$, for some positive integer $k \geq 2$. Let $(G, \mathcal{H})$ be a cross-free graph system of genus g and $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ a 2-coloring such that $d = k$, i.e., the maximum depth of a maximal red vertex is k. We apply the process below to obtain a cross-free system $(G', \mathcal{H}')$ such that the maximum depth of a maximal red vertex is at most $k - 1$, and there are no adjacent twins among the red vertices. We do this in two steps.

In the first step, we apply vertex bypassing to all the vertices in S_k, and we assign color $\boldsymbol{r}$ to all the newly added vertices during the vertex bypassing process. Note that if $v \in S_k$, then by Lemma 2.6 none of the newly added vertices is maximal. By Lemma 2.3, the resulting graph system remains cross-free. In the second step, we repeatedly contract all the adjacent red twins and color the contracted vertex $\boldsymbol{r}$. Clearly, the resulting graph system $(G', \mathcal{H}')$ remains cross-free and does not contain any adjacent red twin.

In $(G', \mathcal{H}')$, each maximal red vertex has depth at most $k - 1$. To see this, note that all the vertices of G in S_k have been bypassed. It is possible that during the edge contractions in the second step above, two non-maximal adjacent red twins x and y may result in a maximal red vertex z. However, at least one of x and y must be added during the vertex bypassing operation since we assumed that G does not contain adjacent red twins. By Lemma 2.6, $\text{DEPTH}(z) \leq k-1$.

Therefore, $(G', \mathcal{H}')$ is a cross-free graph system of genus g such that the maximum depth of a maximal red vertex is at most $k-1$, and there are no adjacent red twins. By the induction hypothesis, $(G', \mathcal{H}')$ admits a primal support Q of genus at most g, which is also a primal support for $(G, \mathcal{H})$ since the set of blue vertices remains unchanged for each subgraph in $\mathcal{H}$. ■

§ 2.5.2 Dual support

In this section, we show that for a cross-free system on a graph of genus g, there is a dual support of genus at most g. For the construction of a dual support, we start with a special case of the problem where it is easy to obtain a dual support. We obtain a support for the general instance by reducing it to this special case by repeatedly applying Vertex Bypassing to a vertex of maximum depth.

An edge $\{u, v\} \in E(G)$ is said to be a *special edge* if $\mathcal{H}_u \neq \emptyset$, $\mathcal{H}_v \neq \emptyset$ and $\mathcal{H}_u \cap \mathcal{H}_v = \emptyset$, i.e., there is no subgraph in $\mathcal{H}$ that contains edge $\{u, v\}$. Let $\text{Spl}_{\mathcal{H}}(E)$ be the set of special edges in $E(G)$ w.r.t. $\mathcal{H}$. A dual support Q^* for $(G, \mathcal{H})$ satisfies the *special edge property* if for each $e = \{u, v\} \in \text{Spl}_{\mathcal{H}}(E)$, there is an edge in Q^* between some $H \in \mathcal{H}_u$ and $H' \in \mathcal{H}_v$.

Lemma 2.8. Let $(G, \mathcal{H})$ be a graph system of genus g such that $\text{DEPTH}(v) \leq 1$ for each $v \in V(G)$. Then, there is a dual support Q^* of genus g on $\mathcal{H}$ with the special edge property.

Proof. Each vertex of G has depth at most 1 and therefore, no two subgraphs in $\mathcal{H}$ share a vertex. We repeatedly contract each edge $e = \{u, v\}$ such that $\mathcal{H}_u \subseteq \mathcal{H}_v$ until no such edge remains, breaking ties arbitrarily. Note that this allows us to contract edges incident to a vertex that is not contained in any subgraph in $\mathcal{H}$. Remove multi edges if any, from the graph thus obtained. The resulting graph Q^* has genus g. See Fig. 2.6. We show that Q^* is the desired support.

For each vertex $v \in V(G)$, the subgraphs containing v trivially induce a connected subgraph in Q^* since $\text{DEPTH}(v) \leq 1$ for all $v \in V(G)$. Consider a special edge $e = \{u, v\} \in E(G)$. Since each of the two endpoints belongs to a subgraph in $\mathcal{H}$, and $\mathcal{H}_u \cap \mathcal{H}_v = \emptyset$, it implies that $\mathcal{H}_u \not\subseteq \mathcal{H}_v$ and $\mathcal{H}_v \not\subseteq \mathcal{H}_u$. In the construction of Q^* therefore, e is not contracted. Further, u and v are contracted to the vertices corresponding to the unique subgraph containing u and v, respectively. Therefore, Q^* satisfies the special edge property. ■

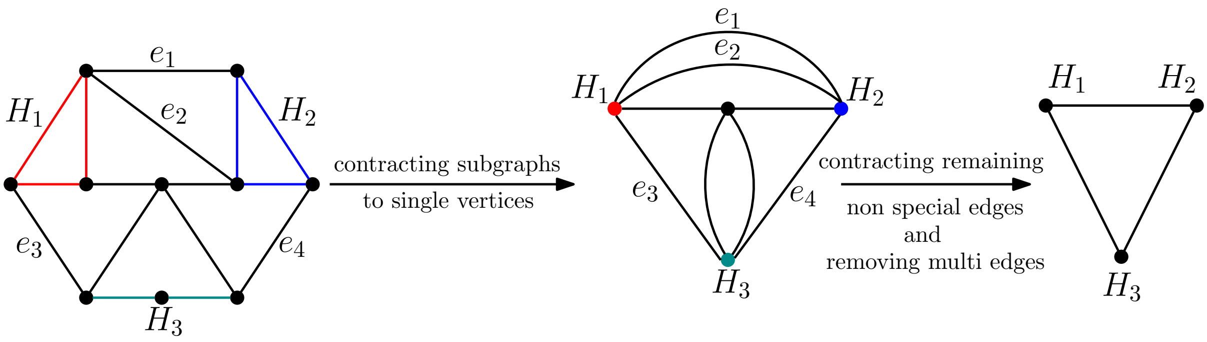


Figure 2.6: Left: a graph G and subgraphs H_1, H_2, H_3 shown red, blue and green respectively. Middle: Subgraphs contracted to single vertices. Right: Dual support with special edge property.

We are now ready to prove the main result of this section. We use induction on the pair (d, n_d) with d being the maximum depth of a vertex, and n_d, the number of vertices of depth d. The base case is derived from Lemma 2.8, while the induction step relies on the operation of vertex bypassing at a vertex of maximum depth. We finish the proof by claiming that the dual support constructed in the inductive step is the required dual support for the input hypergraph.

Theorem 2.4. Let $(G, \mathcal{H})$ be a cross-free system of genus g. Then, there exists a dual support Q^* on $\mathcal{H}$ of genus at most g that satisfies the special edge property.

Proof. By Corollary 2.2, we assume that there are no containments in $\mathcal{H}$. Now consider the embedded cross-free system $(G, \mathcal{H})$. We prove the result by induction on cross-free graph systems lexicographically ordered by (d, n_d), where d is the maximum depth of a vertex in G and n_d is the number of vertices of depth d.

If $d = 1$, then Lemma 2.8 guarantees a support satisfying the special edge property. So suppose $d > 1$ and let v be a vertex of maximum depth, i.e., $\text{DEPTH}(v) = d$. Then, we can assume that v is maximal and thus $\mathcal{H}_e \subsetneq \mathcal{H}_v$ for all $e = \{u, v\} \in E(G)$. Otherwise, if $\mathcal{H}_e = \mathcal{H}_v$ for some edge $e = \{u, v\}$, then $\mathcal{H}_v \subseteq \mathcal{H}_u$ since $\mathcal{H}_e \subseteq \mathcal{H}_u$ for any edge e incident to a vertex u. This implies $\mathcal{H}_u = \mathcal{H}_v$ since v has the maximum depth. Hence, contracting e we obtain a lexicographically smaller graph system $(G/e, \mathcal{H})$, where G/e is the graph obtained from G by contracting e. By the inductive hypothesis, there is a dual support Q^* with the special edge property for the cross-free system $(G/e, \mathcal{H})$. Q^* is also a dual-support for $(G, \mathcal{H})$ since $\mathcal{H}_u = \mathcal{H}_v$. Moreover, Q^* satisfies the special edge property since the contracted edge e is not a special edge.

Therefore, we can assume that for a maximum depth vertex v, and each $e \sim v$, $\mathcal{H}_e \subsetneq \mathcal{H}_v$. We apply $\mathrm{VB}(v)$ to obtain the system $(G', \mathcal{H}')$. By Lemma 2.6, the new vertices $u_0, \ldots, u_{deg(v)-1}$ in G' obtained on applying Vertex Bypassing to v have depth at most $d-1$. Hence, $(d', n_{d'}) \prec (d, n_d)$, where d' and $n_{d'}$ are respectively, the depth of a maximum depth vertex, and the number of vertices of maximum depth in $(G', \mathcal{H}')$. We can assume that $\mathcal{H}'_{u_i} \neq \emptyset$ for any $i \in \{0, \ldots, \deg(v)-1\}$. Otherwise, we can contract the edge $\{u_i, u_{i+1}\}$ (with indices taken $\bmod \deg(v)$) as this does not violate the cross-free condition on $\mathcal{H}'$. Further, there is an injective correspondence between the special edges in G and the special edges in G': For $i = 0, \ldots, deg(v)-1$, if $\{v, v_i\}$ in G is a special edge, the corresponding edge $\{u_i, v_i\}$ is a special edge in G'.

By Lemma 2.2, each subgraph $H \in \mathcal{H}'$ is connected in G', and by Lemma 2.3, $(G', \mathcal{H}')$ is cross-free. By the inductive hypothesis, there is a dual support Q^* for $(G', \mathcal{H}')$ satisfying the special edge property. We show that Q^* is also a support for $(G, \mathcal{H})$.

For each $u \neq v \in V(G)$, it follows from the inductive hypothesis that $\mathcal{H}_u$ induces a connected subgraph of Q^*. It remains to show that $\mathcal{H}_v$ induces a connected subgraph of Q^*. Let C denote the cycle $(u_0, \ldots, u_{\deg(v)-1}, u_0)$ added in $\mathrm{VB}(v)$. Since we assumed (by Corollary 2.2) that $\mathcal{H}$ has no containments, it follows that $\cup_{i=0}^{\deg(v)-1} \mathcal{H}'_{u_i} = \mathcal{H}_v$, as there is no subgraph containing only the vertex v. If none of the edges of C are in $\mathrm{Spl}_{\mathcal{H}'}(E)$, then $\mathcal{H}_v$ is connected since adjacent vertices of C share at least one subgraph and by our assumption on applying $\mathrm{VB}(v)$, $\mathcal{H}'_{u_i} \neq \emptyset$ for any $i = \{0, \ldots, deg(v)-1\}$. On the other hand, if an edge $e = \{u_i, u_{i+1}\}$ (where indices are taken $\bmod \deg(v)$) of C is in $\mathrm{Spl}_{\mathcal{H}'}(E)$, by the inductive hypothesis, since Q^* satisfies the special edge property, at least one subgraph from $\mathcal{H}'_{u_i}$ and one subgraph from $\mathcal{H}'_{u_{i+1}}$ are adjacent in Q^*. Since $\cup_{i=0}^{\deg(v)-1} \mathcal{H}'_{u_i} = \mathcal{H}_v$, and C is a cycle, it follows that $\mathcal{H}_v$ is connected and thus Q^*, is the desired dual support for $(G, \mathcal{H})$ that also satisfies the special edge property. ■

We conclude this section with the following observation. By the special edge property of a dual support, we make use of Observation 2.5.1 below for the construction of an intersection support in the upcoming section.

Observation 2.5.1. *Given a graph system $(G, \mathcal{H})$, let Q^* be a dual support with the special edge property. Let $H_u = \{u\}$ and $H_v = \{v\}$ be subgraphs in $\mathcal{H}$ for some depth 1 vertices $u, v \in V(G)$. Then we can assume without loss of generality that u and v are adjacent in G if and only if H_u and H_v are adjacent in Q^*. Indeed, if u, v are adjacent in G, then H_u, H_v should be adjacent in Q^* by the special edge property. If u, v are not adjacent in G but H_u, H_v are adjacent in Q^*, then removing the edge $\{H_u, H_v\}$ from Q^**

does not violate the conditions of a dual support with the special edge property.

§ 2.5.3 Intersection support

In this section, we show how we can construct an intersection support for a cross-free intersection system $(G, \mathcal{H}, \mathcal{K})$. A vertex $v \in G$ s.t. $\mathcal{H}_v = \emptyset$, but $\mathcal{K}_v \neq \emptyset$ is called a $\mathcal{K}$-vertex, i.e., a vertex of G that is contained in one or more subgraphs in $\mathcal{K}$ but none of the subgraphs in $\mathcal{H}$.

Our proof proceeds in three steps: First, we show that if there are no $\mathcal{K}$-vertices in G, then a dual support for the graph system $(G, \mathcal{H})$ is also an intersection support for $(G, \mathcal{H}, \mathcal{K})$. Secondly, if there are $\mathcal{K}$-vertices in G, then we apply vertex bypassing so that none of the $\mathcal{K}$-vertices are maximal w.r.t. $\mathcal{K}$. We can then add a set of *dummy subgraphs* $\mathcal{F}$ so that there are no $\mathcal{K}$-vertices in $\mathcal{H} \cup \mathcal{F}$. Now, a dual support Q^* for $(G, \mathcal{H} \cup \mathcal{F})$ is an intersection support for $(G, \mathcal{H} \cup \mathcal{F}, \mathcal{K})$ by the earlier argument. Finally, since Q^* is an intersection support, each $K \in \mathcal{K}$ induces a connected subgraph $\mathcal{H}_K$ of Q^*. The collection $\{\mathcal{H}_K\}_{K \in \mathcal{K}}$ of connected subgraphs of Q^* need not be cross-free, however, since no $\mathcal{K}$-vertex was maximal, we can show that no $\mathcal{F}$ vertex in Q^* is maximal. We color the vertices of Q^* as follows: vertices corresponding to $\mathcal{F}$ get colored $\boldsymbol{r}$, and those corresponding to $\mathcal{H}$ get colored $\mathbf{b}$. By appealing to Lemma 2.7, we obtain the desired intersection support $\tilde{Q}$ for $(G, \mathcal{H}, \mathcal{K})$.

In the lemma below, observe that we only require that the subgraphs in $\mathcal{K}$ are connected, and not necessarily cross-free.

Lemma 2.9. Let $(G, \mathcal{H})$ be a cross-free system of genus g and let $\mathcal{K}$ be a set of connected subgraphs of G. If G does not contain a $\mathcal{K}$-vertex, then a dual support Q^* for $(G, \mathcal{H})$ of genus g satisfying the special edge property, is also an intersection support for $(G, \mathcal{H}, \mathcal{K})$.

Proof. By Theorem 2.4, there is a dual support Q^* of genus at most g for the graph system $(G, \mathcal{H})$ with the special edge property. That is, for any vertex $v \in V(G)$, the subgraphs $\mathcal{H}_v$ induce a connected subgraph of Q^*, and for any edge $\{u, v\} \in E(G)$ such that $\mathcal{H}_u \neq \emptyset$, $\mathcal{H}_v \neq \emptyset$, and $\mathcal{H}_{\{u,v\}} = \emptyset$, there is a subgraph $H \in \mathcal{H}_u$ and a subgraph $H' \in \mathcal{H}_v$ such that H and H' are adjacent in Q^*.

For any $K \in \mathcal{K}$, let $H, H' \in \mathcal{H}_K$. We show that there is a path in Q^* between H and H' such that each vertex of this path corresponds to a subgraph in $\mathcal{H}_K$. Let $u \in H \cap K$ and $v \in H' \cap K$.

Since K is connected, there is a path $P = (u = u_0, u_1, \ldots, u_k = v)$ that lies in K. By assumption, none of the vertices of P is a $\mathcal{K}$-vertex. Since Q^* is a dual support for $(G, \mathcal{H})$, for any $i \in \{0, \ldots, k\}$, the subgraphs in $\mathcal{H}_{u_i}$ induce a connected subgraph in

Q^*. For any edge $\{u_i, u_{i+1}\}$, $i = 0, \ldots, k$, either there is a subgraph $H \in \mathcal{H}$ that contains the edge $\{u_i, u_{i+1}\}$, or by the special edge property, there is a subgraph in $\mathcal{H}_{u_i}$ that is adjacent to a subgraph in $\mathcal{H}_{u_{i+1}}$ in Q^*. Since $\mathcal{H}_{u_i} \subseteq \mathcal{H}_K$ for each $i = 0, \ldots, k$, there is a path in Q^* between H and H' consisting only of subgraphs in $\mathcal{H}_K$. Since K was chosen arbitrarily, Q^* is an intersection support for $(G, \mathcal{H}, \mathcal{K})$. ∎

Now, we apply vertex bypassing to obtain a graph system where none of the $\mathcal{K}$-vertices are maximal. For a $\mathcal{K}$-vertex v, if an edge $e \sim v$ is such that $\mathcal{K}_v = \mathcal{K}_e$, we say that e is *full* for v. If a $\mathcal{K}$-vertex does not have a full edge incident on it, then we say that it is maximal w.r.t. $\mathcal{K}$ subgraphs. In this case, $\mathcal{K}_e \subsetneq \mathcal{K}_v$ for all $e \sim v$. Note that a maximum depth $\mathcal{K}$-vertex need not be maximal. In the following, we repeatedly apply vertex bypassing to a maximal $\mathcal{K}$-vertex of maximum depth until no $\mathcal{K}$-vertex is maximal. However, we require an additional property like in the construction of the primal support, that there are no adjacent twins. We recall the definition of twins and adjacent twins here. If two $\mathcal{K}$-vertices u, v of G are contained in the same set of subgraphs of $\mathcal{K}$, they are said to be *twins*. If, in addition, u and v are adjacent, then they are said to be adjacent twins.

The proof of the lemma below follows from arguments similar to those of Theorem 2.3.

Lemma 2.10. Let $(G, \mathcal{H}, \mathcal{K})$ be a cross-free intersection system of genus g. Then, we can modify the arrangement to a cross-free arrangement $(G', \mathcal{H}, \mathcal{K}')$ such that (i) G' has genus g, (ii) no $\mathcal{K}$-vertex in $(G', \mathcal{H}, \mathcal{K}')$ is maximal, (iii) no two $\mathcal{K}$-vertices are adjacent twins, and (iv) an intersection support for $(G', \mathcal{H}, \mathcal{K}')$ is also an intersection support for $(G, \mathcal{H}, \mathcal{K})$.

Proof. We assume that a cross-free embedding of $(G, \mathcal{H}, \mathcal{K})$ is given, and with a slight abuse of notation, we use G to also refer to the embedded graph. We will modify the cross-free graph system $(G, \mathcal{H}, \mathcal{K})$ to a cross-free graph system $(G', \mathcal{H}, \mathcal{K}')$ such that G' has an embedding on the same surface as G does, and that the intersection hypergraph defined by $(G, \mathcal{H}, \mathcal{K})$ is isomorphic to that defined by $(G', \mathcal{H}, \mathcal{K}')$. This will prove (i) and (iv). Also, in the process of obtaining $(G', \mathcal{H}, \mathcal{K}')$, as long as there are adjacent twins among $\mathcal{K}$-vertices, we contract them to a single vertex. Thus, we will be done with condition (iii) as well. So, in the rest of the proof, we claim to establish condition (ii) of the lemma.

Let d denote the maximum depth of a maximal $\mathcal{K}$-vertex in G. We prove the argument by induction on d. For $i = 1, 2, \ldots, d$, let S_i denote the set of maximal $\mathcal{K}$-vertices of depth i. Suppose $d = 1$, and consider any $v \in S_1$. By the definition of a maximal vertex w.r.t. $\mathcal{K}$, the subgraph $K \in \mathcal{K}$ containing v, does not contain any other vertex of G. Thus $\mathcal{H}_K = \emptyset$. Let $\mathcal{K}_1 \subseteq \mathcal{K}$ denote the collection of all subgraphs consisting of single vertices in S_1, and let $\mathcal{K}' = \mathcal{K} \setminus \mathcal{K}_1$. Then $(G', \mathcal{H}, \mathcal{K}')$ is the required graph system, where $G' = G$.

Let our claim be true for $d < k$ for some positive integer $k \geq 2$. Consider a cross-free graph system $(G, \mathcal{H}, \mathcal{K})$ and a 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ such that $d = k$. We apply vertex bypassing to all the vertices in S_k. Note that this does not modify any subgraph in $\mathcal{H}$. For any pair of adjacent twins created among the $\mathcal{K}$-vertices, we contract them to a single $\mathcal{K}$-vertex until there is no such twin. Let $(G'', \mathcal{H}, \mathcal{K}'')$ denote the resulting graph system. By Lemma 2.3 and the contraction among the adjacent $\mathcal{K}$-twins, it follows that $(G'', \mathcal{K}'')$ remains cross-free. Since for any vertex $v \in V(G)$ that lies in a subgraph $H \in \mathcal{H}$, the cyclic order of its neighbours is preserved in G'', the graph system $(G'', \mathcal{H})$ is cross-free. Therefore, $(G'', \mathcal{H}, \mathcal{K}'')$ is a cross-free intersection system. By the construction of G'', note that a subgraph $H \in \mathcal{H}$ intersects a subgraph $K \in \mathcal{K}$ in G if and only if H intersects in G'' with the representative K'' of K. It follows that the intersection hypergraphs defined by the two intersection systems are isomorphic.

By an argument similar to that in the proof of Theorem 2.3, the maximum depth of a maximal $\mathcal{K}$-vertex in $(G'', \mathcal{H}, \mathcal{K}'')$ is at most $k - 1$. By the induction hypothesis, $(G'', \mathcal{H}, \mathcal{K}'')$ can be modified to a graph system $(G', \mathcal{H}, \mathcal{K}')$ satisfying properties (i)-(iv) of the lemma. $(G', \mathcal{H}, \mathcal{K}')$ is the required cross-free intersection system since the intersection hypergraph defined by $(G', \mathcal{H}, \mathcal{K}')$ is isomorphic to that defined by $(G'', \mathcal{H}, \mathcal{K}'')$, which itself is isomorphic to the intersection hypergraph defined by $(G, \mathcal{H}, \mathcal{K})$. ■

By Lemma 2.10, it is sufficient to prove an intersection support for $(G, \mathcal{H}, \mathcal{K})$ when no $\mathcal{K}$-vertex is maximal, and there is no adjacent twin among $\mathcal{K}$-vertices. At each $\mathcal{K}$-vertex v, we add a *dummy subgraph* F_v. Let $\mathcal{F}$ denote the set of dummy subgraphs thus added. For the intersection system $(G, \mathcal{H} \cup \mathcal{F}, \mathcal{K})$, we can obtain an intersection support Q by applying Lemma 2.9. We now show that we can obtain an intersection support for $(G, \mathcal{H}, \mathcal{K})$ from Q using Lemma 2.7 and Observation 2.5.1.

Theorem 2.5. Let $(G, \mathcal{H}, \mathcal{K})$ be a cross-free intersection system of genus g. Then, there exists an intersection support $\tilde{Q}$ on $\mathcal{H}$ of genus at most g.

Proof. If G does not contain any $\mathcal{K}$-vertex, then by Lemma 2.9, we are done. So, suppose that there are some $\mathcal{K}$-vertices in G. By Lemma 2.10, we can assume without loss of generality that no $\mathcal{K}$-vertex is maximal, and that there are no adjacent twins among the $\mathcal{K}$-vertices.

At each $\mathcal{K}$-vertex v of G, we add a dummy subgraph F_v. Let $\mathcal{F}$ denote the set of dummy subgraphs added. Let $\mathcal{H}' = \mathcal{H} \cup \mathcal{F}$ and consider the intersection system

$(G, \mathcal{H}', \mathcal{K})$. By Lemma 2.9, there is a dual support Q for $(G, \mathcal{H}')$ with the special edge property such that Q is an intersection support for $(G, \mathcal{H}', \mathcal{K})$, and Q has genus at most g. Since Q is an intersection support, each $K \in \mathcal{K}$ induces a connected subgraph $\mathcal{H}'_K$ of Q. Abusing notation, we use the labels of the subgraphs in $\mathcal{H} \cup \mathcal{F}$ to denote their corresponding vertices in Q. Consider a dummy subgraph F_u corresponding to a $\mathcal{K}$-vertex u. Since u is not maximal in G w.r.t. $\mathcal{K}$, it has a full edge $e = \{u, v\}$ incident to it, i.e., $\mathcal{K}_u = \mathcal{K}_e$. But, this implies $\mathcal{K}_u \subseteq \mathcal{K}_v$. Note that for every edge incident to u in G, there is a corresponding edge in the dual support Q as a special edge incident to the dummy subgraph F_u, since F_u consists of a single vertex u, and $|\mathcal{H}'_u| = 1$. In particular, let e' be the copy of e incident to F_u in Q. If $X \in \mathcal{H} \cup \mathcal{F}$ is the other end of e', then $\mathcal{K}_{F_u} \subseteq \mathcal{K}_X$ in the graph system $(Q, \mathcal{K})$, i.e., F_u is not a maximal vertex in $(Q, \mathcal{K})$. Hence, none of the dummy subgraphs $\mathcal{F}$ corresponds to a maximal vertex in $(Q, \mathcal{K})$. See Fig. 2.7.

Now we assign a 2-coloring to the vertices of Q. Let $c : V(Q) \to \{\mathbf{b}, \boldsymbol{r}\}$ be such that $c(F) = \boldsymbol{r}$ for each $F \in \mathcal{F}$, and $c(H) = \mathbf{b}$ for each $H \in \mathcal{H}$. To complete the argument, we need to show that there are no adjacent red twins in the graph system $(Q, \mathcal{K})$. Consider any two red vertices F_u and F_v of Q, corresponding to some pair of vertices u and v of G, respectively. Since $|\mathcal{H}_u| = 1 = |\mathcal{H}_v|$, by Observation 2.5.1, we can assume that F_u and F_v are adjacent in Q if and only if u and v are adjacent in G. If F_u and F_v are not adjacent, then trivially they are not adjacent red twins. So, suppose they are adjacent. Then u and v are also adjacent in G. By assumption, G does not contain adjacent $\mathcal{K}$-vertices that are twins, thus $\mathcal{K}_u \neq \mathcal{K}_v$. It follows that in the graph system $(Q, \mathcal{K})$, we have $\mathcal{K}_{F_u} \neq \mathcal{K}_{F_v}$, and hence, F_u and F_v are not twins in Q. Therefore, Q satisfies the conditions of Lemma 2.7 that no *red* vertex is maximal and each $K \in \mathcal{K}$ induces a connected subgraph of Q. Therefore, the graph system $(Q, \mathcal{K})$ admits a support $\tilde{Q}$ on $\mathcal{H}$ of genus at most g. Since each $K \in \mathcal{K}$ induces a connected subgraph in $\tilde{Q}$, we have the desired intersection support for $(G, \mathcal{H}, \mathcal{K})$. ■

§ 2.6 Concluding Remarks and Open Questions

In this chapter, we studied cross-free graph systems. We proved that if $(G, \mathcal{H})$ (similarly, $(G, \mathcal{H}, \mathcal{K})$) is a cross-free system of genus g, there are primal and dual (similarly, intersection) supports of genus at most g. We also proved that non-piercing is not a sufficient condition for the existence of a support of bounded genus, but cross-free is.

▷ Limitations.

Although the results obtained in this chapter advance our understanding of the rela-

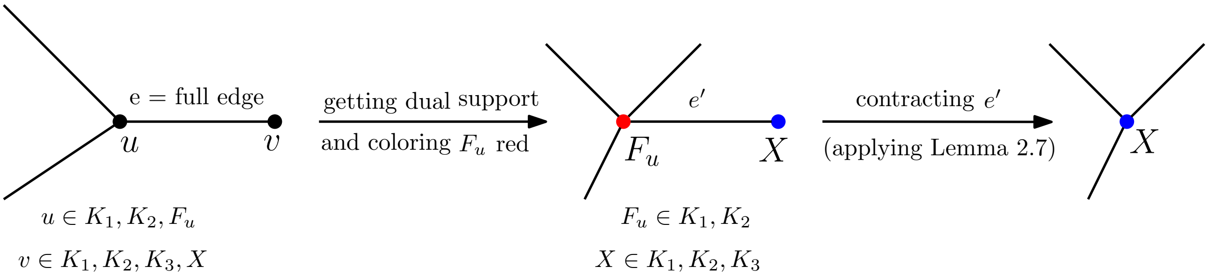


Figure 2.7: Left: a description of intersection system $(G, \mathcal{H}', \mathcal{K})$, where u is not a maximal vertex w.r.t. $\mathcal{K}$. Middle: a dual support Q for $(G, \mathcal{H}')$ with special edge property. F_u is not maximal w.r.t. $(Q, \mathcal{K})$. Right: getting rid of red vertices to obtain in intersection support for $(G, \mathcal{H}, \mathcal{K})$.

tion between a particular topological embedding of the host graph and a support of bounded genus, they are subject to certain gaps. Notably, as we saw in Section 2.3 that non-piercing is not a sufficient condition to obtain a dual or intersection support of bounded genus, but we don't know this for the primal setting. However, as we proved, cross-free is sufficient for the construction of an intersection support of bounded genus. Is this necessary? We believe this to be the case and discuss it further in Section 6.3. A precise characterization of the hypergraphs that admit supports of bounded genus, however, remains out of reach.

Furthermore, we could not demonstrate whether our algorithms for constructing supports of bounded genus run in polynomial time, even when a cross-free embedding of the host graph is provided. The main difficulty is that in the process of vertex bypassing at a vertex v, we add additional vertices while subdividing the edges incident to v. These newly added vertices may require bypassing operations in future steps, and thus, the size of the host graph can become exponentially large. In [RR20], Raman and Ray dealt with a similar difficulty in constructing a support for non-piercing regions in the plane. In their setting, a *cell-bypassing* operation creates additional cells in the arrangement. They managed to overcome it by showing additional structural restrictions on the regions using which they could define a suitable potential function that was polynomially bounded at the start, and decreased by at least 1 in each *cell-bypassing* operation. In our case, however, we were unable to find a suitable potential function to polynomially bound the running time of our algorithm, and we leave it as an intriguing open problem.

Open Problem 1. Let G be an n-vertex graph of bounded genus and $c : V(G) :\rightarrow \{\mathbf{b}, \boldsymbol{r}\}$ be a 2-coloring of the vertices of G. Let $\mathcal{H}$ and $\mathcal{K}$ be two families of subgraphs of G with $|\mathcal{H}| = m_1, |\mathcal{K}| = m_2$ such that $(G, \mathcal{H})$ and $(G, \mathcal{K})$ are simultaneously cross-free in some embedding of G. Do the algorithms to compute primal, dual, or intersection supports, as established in Theorems 2.3 to 2.5, run in time poly(n, m_1, m_2)? If not, are there any other efficient algorithms to construct supports of bounded genus?

The following question may be of independent interest. In Example 2.1, we showed that non-piercing is not a sufficient condition for the construction of a support of bounded genus. However, we are not aware of any such construction for the primal setting.

Open Problem 2. Give a non-piercing graph system $(G, \mathcal{H})$ of genus g with a 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. Does a primal support of genus g exist?

CHAPTER 3

Outerplanar Support

Abstract

It was shown by Buchin et al. (*On planar support for hypergraphs*) that it is NP-hard to decide if a given hypergraph admits a support that is k-outerplanar for any $k \geq 2$, leaving the outerplanar case open. In this chapter, we show that if a hypergraph is derived from non-piercing subgraphs of an outerplanar graph G, then it admits primal, dual, and intersection supports. The algorithms are constructive and run in time poly(n) where n is the number of vertices in G. The content of this chapter is based on the following paper.
"**Supports for Outerplanar and Bounded Treewidth Graphs**".

TABLE OF CONTENTS

§ 3.1 Preliminaries and Notations

In this section, we state the necessary definitions that we will be using throughout this chapter. Recall from Chapter 1 the definition of an outerplanar graph defined below.

> **Definition 3.1** (Outerplanar graph)**.** A graph G is called outerplanar if there is an embedding of G in the plane such that all its vertices lie on the exterior face.

Throughout this chapter, G will always denote an outerplanar graph unless otherwise stated. Examples of graphs that are outerplanar include forests and cycles. However, the complete graph K_4 and the complete bipartite graph $K_{2,3}$ are *not* outerplanar. It is well known that a graph is outerplanar if and only if it does not contain a subgraph homeomorphic to K_4 or $K_{2,3}$ [CH67].

Let G be an outerplanar graph with a 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ of $V(G)$. Let $\mathcal{H}, \mathcal{K}$ be any two collections of subgraphs of G. As in the previous chapter, the tuples $(G, \mathcal{H})$ and $(G, \mathcal{H}, \mathcal{K})$ will be called graph systems and intersection systems, respectively. As a special case, if the host graph is a cycle C, then we will call $(C, \mathcal{H})$ a *cycle system* and $(C, \mathcal{H}, \mathcal{K})$ a *cycle intersection system*.

Recall Definition 2.5 of non-piercing subgraphs of a graph. Adding edges to G does not violate the non-piercing property of its subgraphs. So, we assume throughout this chapter that the outerplanar graph G is *edge-maximal* so that there is a cycle C in an outerplanar embedding of G. It is clear that a primal or dual hypergraph defined by $(G, \mathcal{H})$ is the same as that defined by $(C, \mathcal{H}')$ where $\mathcal{H}'$ denotes the restriction of $\mathcal{H}$ into C, i.e., for each $H \in \mathcal{H}$, there is $H' \in \mathcal{H}'$ such that $V(H) = V(H')$, and vice-versa. So, we will be using $(C, \mathcal{H})$ instead of $(C, \mathcal{H}')$ to define a cycle system arising from an outerplanar graph system $(G, \mathcal{H})$. A similar argument holds for an intersection hypergraph defined by $(G, \mathcal{H}, \mathcal{K})$, hence the following remark.

> **Remark 3.1.1.** To construct an intersection support for a non-piercing outerplanar system $(G, \mathcal{H}, \mathcal{K})$, it is sufficient to construct a support for the corresponding cycle system $(C, \mathcal{H}, \mathcal{K})$ where C is the outer cycle in an outerplanar embedding of G.

Since intersection hypergraphs generalize primal and dual hypergraphs, the remark above is also applicable to the primal and dual settings.

If $\mathcal{H}$ and $\mathcal{K}$ are non-piercing subgraphs of an outerplanar graph G, then the corresponding cycle system is *(strong)* $axax$*-free*, a notion we define below.

Definition 3.2 ($axax$-free)**.** Let $(C, \mathcal{H})$ be a cycle system. $H, H' \in \mathcal{H}$ are an $axax$-pair if there are four distinct vertices a_1, x_1, a_2, x_2 in cyclic order on C such that $a_1, a_2 \in H \setminus H'$ and $x_1, x_2 \in H'$. $(C, \mathcal{H})$ is $axax$-free if there are no $axax$-pairs in $\mathcal{H}$.

For two families $\mathcal{H}, \mathcal{K}$ of subgraphs of C, the intersection system $(C, \mathcal{H}, \mathcal{K})$ is said to satisfy the *$axax$-free property* if both $(C, \mathcal{H})$ and $(C, \mathcal{K})$ are $axax$-free. Let $H \in \mathcal{H}$ and $K \in \mathcal{K}$. If for a sequence of four vertices h_1, k_1, h_2, k_2 in this order on C with $h_1, h_2 \in H$ and $k_1, k_2 \in K$ it implies that $H \cap K \neq \emptyset$, then we say that the pair (H, K) satisfies satisfy the *intersection property*. If there is no such sequence of four vertices, then (H, K) trivially satisfies the intersection property. Finally, $(C, \mathcal{H}, \mathcal{K})$ is said to satisfy the *intersection property* if it is satisfied by the pair (H, K) for all $H \in \mathcal{H}$ and $K \in \mathcal{K}$.

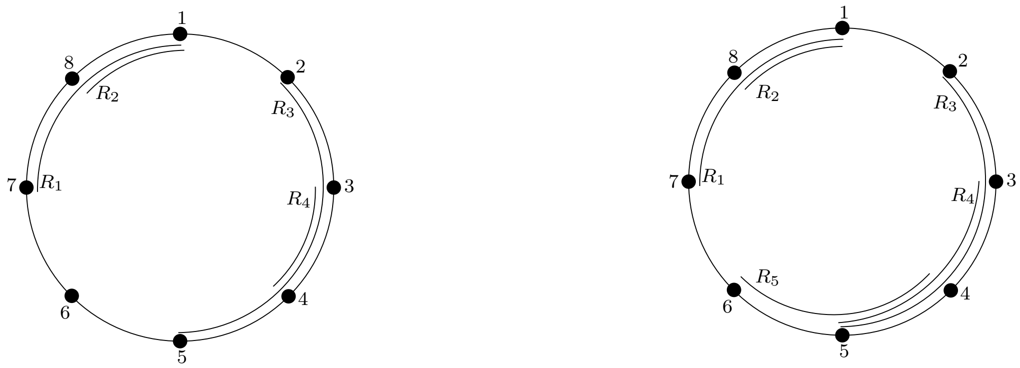
Definition 3.3 ((Strong) $axax$-free)**.** The intersection system $(C, \mathcal{H}, \mathcal{K})$ is *strong $axax$-free property* if $(C, \mathcal{H}, \mathcal{K})$ satisfies the $axax$-free property and the intersection property.

(a) $axax$-free **(b)** Strong $axax$-free

Figure 3.1: Figures above shows subgraphs $\mathcal{H} = \{H_1, H_2, H_3\}$, and $\mathcal{K} = \{K\}$. Each vertex is labelled with the subgraph containing it. In (a), $\mathcal{H}$ is $axax$-free, and in (b), $(C, \mathcal{H}, \mathcal{K})$ is strong $axax$-free where C is the cycle.

Figure Fig. 3.1 shows examples of $axax$-free and strong $axax$-free cycle systems. Now, we state some basic terminology required to prove our main results. Let $C = \{0, \ldots, n-1\}$ be a cycle on n vertices oriented clockwise, and let $\mathcal{R}$ be a collection of arcs on C whose both ends are defined by the vertices of C.

For an $R \in \mathcal{R}$, if $R = \mathrm{arc}[i, j]$, i.e., R consists of a consecutive sequence of vertices $[i, i+1, \ldots, j]$ where the indices are numbered $\pmod{n}$, we say that R is a *run* on C. We also use $\mathrm{arc}(i, j)$ to denote the run $[i+1, \ldots, j-1]$. Let $s(R) = i$ and $t(R) = j$. Consider a pair of arcs $R, R' \in \mathcal{R}$ such that $R \subseteq R'$. In a traversal of C starting at $s(R')$, if we have $s(R') < s(R) \leq t(R) < t(R')$, then we say that R is *strictly contained* in R'. If $s(R) = s(R')$, or $t(R) = t(R')$, then we say that R is *weakly contained* in R'. If there is no pair of arcs $R, R' \in \mathcal{R}$ such that R is strictly contained in R', then we say that $\mathcal{R}$ is *strict-containment free*. If there is no pair of arcs $R, R' \in \mathcal{R}$ that are weakly contained in each other or strictly contained in each other, then $\mathcal{R}$ is *containment-free*, i.e., $R \setminus R' \neq \emptyset$ for all $R, R' \in \mathcal{R}$. See Fig. 3.2.



(a) R_2 is weakly contained in R_1, and R_4 is strictly contained in R_3.

(b) Strict-containment free runs on the cycle

Figure 3.2: Weak containment, strict containment and strict-containment free. Induced subgraphs of the cycle are shown by corresponding arcs inside the cycle.

Strict containment defines a natural partial order on $\mathcal{R}$. Let $\mathcal{R}^* \subseteq \mathcal{R}$ denote the maximal elements of this *strict-containment order*. Then, $\mathcal{R}^*$ is a maximal strict-containment free subset of $\mathcal{R}$. Note that $\mathcal{R}^*$ can contain subgraphs that are weakly contained in each other.

Let $\mathcal{H}$ and $\mathcal{K}$ be any collections of subgraphs of a cycle C. Then in the cycle system $(C, \mathcal{H}, \mathcal{K})$, each subgraph $X \in \mathcal{H} \cup \mathcal{K}$ induces a sequence of runs on C. Let n_X denote the number of runs of X in C. Let $r_0(X) = [s_0, \ldots, t_0], r_1(X) = [s_1, \ldots, t_1], \ldots, r_{k-1} = [s_{k-1}, \ldots, t_{k-1}]$ denote the $k = n_X$ runs of X on C. For $i = 0, \ldots, k-1$, let d_i denote the *chord* $\{t_i, s_{i+1}\}$. Let $\ell(d_i) = |\mathrm{arc}[t_i, s_{i+1}]|$, and let $\ell(X) = \arg\min_{d_i} \ell(d_i)$, with ties broken arbitrarily. If $n_X = 1$, then d_X is undefined.

§ 3.2 Basic Tools for Cycle $axax$-free Systems

We start with the use of Lemma 2.5 in the context of outerplanar graphs. If G' is a graph obtained from an outerplanar graph G by adding a degree-one vertex to G, then G' is also outerplanar. We state below a corollary that is a direct implication of Lemma 2.5.

Corollary 3.1. Let $(G, \mathcal{H}, \mathcal{K})$ be an intersection system where G is an outerplanar graph. In order to construct a dual support for $(G, \mathcal{H})$ or an intersection support for $(G, \mathcal{H}, \mathcal{K})$ that is outerplanar, it is sufficient to construct such a support when $\mathcal{H}$ is containment-free.

Now show that the cycle system obtained from an outerplanar non-piercing system is strong $axax$-free.

Lemma 3.1. Let $(G, \mathcal{H}, \mathcal{K})$ be an embedded outerplanar non-piercing system with C denoting the outer cycle of G. Then, $(C, \mathcal{H}, \mathcal{K})$ satisfies the strong $axax$-free property.

Proof. Let $H, H' \in \mathcal{H}$. Since $\mathcal{H}$ are non-piercing subgraphs of G, $H \setminus H'$ and $H' \setminus H$ are both connected. Suppose there is a cyclic sequence a_1, x_1, a_2, x_2 of the vertices on C such that $a_1, a_2 \in H \setminus H'$, and $x_1, x_2 \in H'$. See Fig. 3.3. Since $H \setminus H'$ is connected, there is a path P between a_1 and a_2 lying in $H \setminus H'$. Then, x_1 and x_2 lie in separate components of $G \setminus P$, contradicting the assumption that $H' \setminus H$ is connected. An identical argument shows that $(C, \mathcal{K})$ is $axax$-free.

To see that $(C, \mathcal{H}, \mathcal{K})$ satisfies the intersection property, consider $H \in \mathcal{H}$ and $K \in \mathcal{K}$ with vertices h_1, k_1, h_2, k_2 in cyclic order s.t. $h_1, h_2 \in H$ and $k_1, k_2 \in K$. Since H is connected in G, there is a path P between h_1 and h_2, all of whose vertices lie in H. If $H \cap K = \emptyset$, then P lies in $H \setminus K$. Since h_1 and h_2 are non-consecutive on C, $G \setminus P$ gets separated into two components with k_1 and k_2 in distinct components. This implies there is no path between k_1 and k_2 in G that lies entirely in K, a contradiction since K is connected. ■

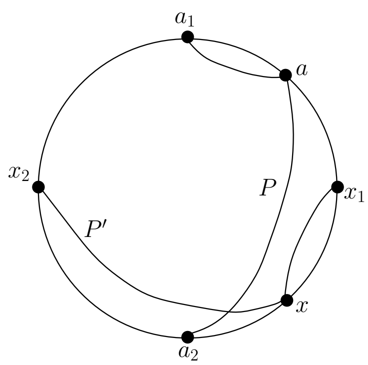


Figure 3.3: Non-piercing implies $axax$-free.

A consequence of a cycle system $(C, \mathcal{H}, \mathcal{K})$ being $axax$-free that will be useful later is the following proposition which is a slight variation of Corollary 3.1. The proof is along the same lines as Lemma 2.5.

Proposition 3.1. Let $(C, \mathcal{H}, \mathcal{K})$ be a cycle $axax$-free system where subgraphs in $\mathcal{H}$ are single runs, and let $\mathcal{H}^*$ be the maximal elements in the strict-containment order on $\mathcal{H}$. If $(C, \mathcal{H}^*, \mathcal{K})$ admits an outerplanar support, then so does $(C, \mathcal{H}, \mathcal{K})$.

We now show that we can simplify an $axax$-free cycle system so that each vertex of C is contained in at least one subgraph in $\mathcal{H}$ and one subgraph in $\mathcal{K}$.

Definition 3.4 (Reduced cycle system)**.** A cycle system $(C, \mathcal{H}, \mathcal{K})$ is called *reduced* if for each vertex v of C, there is an $H \in \mathcal{H}$ and a $K \in \mathcal{K}$ such that $v \in V(H) \cap V(K)$.

Consider a cycle system $(C, \mathcal{H}, \mathcal{K})$. Let C' be the induced cycle obtained from C by removing each vertex v of C and making its neighbors adjacent whenever no subgraph in $\mathcal{H}$ contains v, or no subgraph in $\mathcal{K}$ contains v. Let $\mathcal{H}' = \{H \cap C' : H \in \mathcal{H}\}$ and $\mathcal{K}' = \{K \cap C' : K \in \mathcal{K}\}$. It follows that $(C', \mathcal{H}', \mathcal{K}')$ is a reduced cycle system. The following proposition shows that it is sufficient to construct a support when $(C, \mathcal{H}, \mathcal{K})$ is a reduced system.

Proposition 3.2. If $(C, \mathcal{H}, \mathcal{K})$ is a strong $axax$-free system, then the reduced system $(C', \mathcal{H}', \mathcal{K}')$ is also a strong $axax$-free system. Further, any intersection support for $(C', \mathcal{H}', \mathcal{K}')$ is also an intersection support for $(C, \mathcal{H}, \mathcal{K})$.

Proof. If $(C, \mathcal{H}, \mathcal{K})$ is strong $axax$-free, removing an $\mathcal{H}$-vertex or a $\mathcal{K}$-vertex leaves the reduced system $(C', \mathcal{H}', \mathcal{K}')$ is strong $axax$-free and the resulting intersection hypergraph remains the same since we did not remove any vertex $v \in H \cap K$ for any $H \in \mathcal{H}$ and $K \in \mathcal{K}$. Hence, a support for the reduced system is also a support for the original system. ∎

As a consequence of Proposition 3.2, we assume throughout throughout the rest of this chapter that $(C, \mathcal{H}, \mathcal{K})$ is a reduced system, and by Corollary 3.1, or Proposition 3.1, we assume that $\mathcal{H}$ is either containment-free, or it is strict-containment free (as will be useful in our proofs).

§ 3.3 Construction of Outerplanar Supports

In this section, we prove our main result - the construction of an outerplanar support for the intersection hypergraph. Before we proceed, we keep in mind Fig. 1.5 and Remark 2.3.1 which illustrate support in primal, dual, and intersection hypergraphs, as well as the notions of cross-free and non-piercing subgraphs.

We show that for a non-piercing outerplanar system $(G, \mathcal{H}, \mathcal{K})$, there is an outerplanar support. To show this, it is enough to provide an intersection support for the corresponding cycle system $(C, \mathcal{H}, \mathcal{K})$, where C is the outer cycle in an outerplanar embedding of G.

Theorem 3.2. Let $(C, \mathcal{H}, \mathcal{K})$ be a strong $axax$-free outerplanar system. Then, there is an outerplanar support Q for $(C, \mathcal{H}, \mathcal{K})$.

By Lemma 3.1 and Theorem 3.2 in hand, the following theorem is immediate, which is the central result of this chapter.

Theorem 3.3. If $(G, \mathcal{H}, \mathcal{K})$ is an outerplanar non-piercing system, then there is an intersection support for $(G, \mathcal{H}, \mathcal{K})$ which is an outerplanar graph.

Proof. Let C be the outer cycle in an outerplanar embedding of G. By Lemma 3.1, $(C, \mathcal{H}, \mathcal{K})$ is strong $axax$-free. By Theorem 3.2, there is an outerplanar support $\tilde{Q}$ for $(C, \mathcal{H}, \mathcal{K})$. $\tilde{Q}$ is also a support for $(G, \mathcal{H}, \mathcal{K})$ since the underlying intersection hypergraph defined by $\mathcal{H}$ and $\mathcal{K}$ remains the same on G and on C. ■

The existence of an intersection support implies the existence of a primal and a dual support. For completeness, we state the following corollaries.

Corollary 3.4. Let $(G, \mathcal{H})$ be an outerplanar non-piercing graph system and $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. Then, there is a primal support that is outerplanar.

Proof. Consider an intersection system $(G, \mathcal{H}', \mathcal{H})$ where subgraphs in $\mathcal{H}'$ consists of blue vertices of G, i.e., $\mathcal{H}' = \{\{v\} : v \in V(G) \textit{ and } c(v) = \mathbf{b}\}\}$. Then $(G, \mathcal{H}', \mathcal{H})$ is a non-piercing system and thus has an intersection support Q that is outerplanar. It is easy to see that Q is also a primal support for $(G, \mathcal{H})$. ■

Corollary 3.5. Let $(G, \mathcal{H})$ be an outerplanar non-piercing system. Then there is a dual support that is outerplanar.

Proof. Consider an intersection system $(G, \mathcal{H}, \mathcal{K})$ where $\mathcal{K}$ consists of a all the singleton subgraphs of G, i.e., $\mathcal{K} = \{\{v\} : v \in V(G)\}$. Then $(G, \mathcal{H}, \mathcal{K})$ is a non-piercing system and thus admits an outerplanar intersection support Q. By construction, Q is also a dual support for $(G, \mathcal{H})$. ■

In order to complete the proof of Theorem 3.3, we are only left to prove Theorem 3.2, which we do in the rest of this section. Our proof proceeds in *three easy steps*. We start with the case when all subgraphs in $\mathcal{H} \cup \mathcal{K}$ induce single runs on C. We next use this result to obtain an outerplanar support for the case when only the subgraphs in $\mathcal{H}$ are required to induce single runs on C. Finally, we use the outerplanar supports guaranteed by the two special cases to obtain an outerplanar support for the general case.

Let $\mathcal{R}$ be a collection of arcs on a cycle C that are strict-containment free. Starting at an arbitrary vertex on C, in a clockwise traversal, we construct a cycle $C(\mathcal{R})$ on $\mathcal{R}$ as follows: When we visit a vertex v of C, for each $R \in \mathcal{R}$ s.t. $s(R) = v$, we put a vertex in $C(\mathcal{R})$. The vertices added to $C(\mathcal{R})$ w.r.t. v are ordered in increasing order of $t(R)$, with ties broken arbitrarily. We call the cycle $C(\mathcal{R})$ thus constructed, the cycle in *lex. cyclic order*, and we say that $\mathcal{R}$ is ordered in *lex. cyclic order*.

Starting with a simpler case, let each subgraph in $\mathcal{H} \cup \mathcal{K}$ induce a single run on C. For the requirement in the general settings, we assume by Proposition 3.1 that $\mathcal{H}$ is strict-containment free. We claim that $C(\mathcal{H})$, the cycle on $\mathcal{H}$ in lex. cyclic order is the desired support for $(C, \mathcal{H}, \mathcal{K})$.

Lemma 3.2. Let $(C, \mathcal{H}, \mathcal{K})$ be a cycle system such that $\mathcal{H}$ is strict-containment free and each $X \in \mathcal{H} \cup \mathcal{K}$ induce a single run on C. Then, the cycle $C(\mathcal{H})$ is an outerplanar support for $(C, \mathcal{H}, \mathcal{K})$.

Proof. $C(\mathcal{H})$ is clearly an outerplanar graph. Consider a subgraph $K \in \mathcal{K}$. Since K induces a single run on C and the subgraphs in $\mathcal{H}$ are strict-containment free, it follows that the subgraphs in $\mathcal{H}_K$ appear consecutively in lex. cyclic order on $C(\mathcal{H})$ and thus induce a connected subgraph of $C(\mathcal{H})$. ■

Now, assume that subgraphs in $\mathcal{H}$ induce single runs on C, and those in $\mathcal{K}$ can have multiple runs. We claim that there is an outerplanar support with outer cycle $C(\mathcal{H})$. But, before we do that, we show the following simple consequence of $(C, \mathcal{H}, \mathcal{K})$ being $axax$-free.

Proposition 3.3. Let $(C, \mathcal{H})$ be $axax$-free. For an $H \in \mathcal{H}$, let u, v be two non-consecutive vertices on C with $u, v \in H$. Then, for any $H' \in \mathcal{H}$ such that $H' \cap \mathrm{arc}(u, v) \neq \emptyset \neq H' \cap \mathrm{arc}(v, u)$, H' must contain either u or v. Moreover, if $H \cap \mathrm{arc}(u, v) = \emptyset$, then $H' \cap \mathrm{arc}[v, u] \subseteq H \cap \mathrm{arc}[v, u]$.

Proof. Let x and y be any two vertices of H' in $\mathrm{arc}(u, v)$ and $\mathrm{arc}(v, u)$ respectively, and let $u, v \in H \setminus H'$. The vertices u, x, v, y form an $axax$-pattern on C; a contradiction since $(C, \mathcal{H})$ is $axax$-free.

The proof for the *moreover* part is similar. Indeed, let $h' \in \mathrm{arc}[v, u]$ s.t. $h' \in H' \setminus H$. Also, note that $x \in H' \setminus H$. Hence, the subgraphs H, H' form an $axax$-pattern as witnessed by the cyclic sequence h', u, x, v; a contradiction. ■

Now we prove an outerplanar intersection support when $\mathcal{H}$ is containment-free, and consists of single runs only. We use an induction argument based on the total number of runs in the elements of $\mathcal{K}$. The base case of the induction will be followed from Lemma 3.2, while for the induction step, we chose a chord of minimum length to partition the input cycle into two smaller cycles. The proof will then be guaranteed by the applications of Lemma 3.2 in combination with Proposition 3.3.

Lemma 3.3. Let $(C, \mathcal{H}, \mathcal{K})$ be any $axax$-free cycle system such that each $H \in \mathcal{H}$ induces a single run on C. If $\mathcal{H}$ is strict-containment free, then there exists an outerplanar support for $(C, \mathcal{H}, \mathcal{K})$ with outer cycle $C(\mathcal{H})$.

Proof. Let $N = N(C, \mathcal{K}) = \sum_{K \in \mathcal{K}}(n_K - 1)$, where n_K is the number of runs of K on C. We proceed by induction on N. If $N = 0$, then $n_K = 1$ for all $K \in \mathcal{K}$ and we are done by Lemma 3.2. So, suppose $N \geq 1$, i.e., $n_K \geq 2$ for some $K \in \mathcal{K}$.

We assume the lemma holds when $N(C, \mathcal{K}') < N$ for any $axax$-free system $(C, \mathcal{H}, \mathcal{K}')$. Let $(C, \mathcal{H}, \mathcal{K})$ be an $axax$-free system with $N(C, \mathcal{K}) = N$. Let $K_0 = \arg\min_{K \in \mathcal{K}} \ell(K)$. Let $d_{K_0} = \{u_0, v_0\}$ be the chord of K_0 realizing this minimum. We use the chord d_{K_0} to split C into two cycles.

Let C_L denote the cycle $\mathrm{arc}[v_0, u_0] \cup d_{K_0}$, and C_R denote the cycle $\mathrm{arc}[u_0, v_0] \cup d_{K_0}$. Let $\mathcal{K}_X = \{K \cap C_X : K \in \mathcal{K}\}$, where $X \in \{L, R\}$. Observe that K_0 appears both in C_L and C_R, where in C_R, K_0 spans the vertices $\{u_0, v_0\}$.

Let $H_1, \ldots, H_r$ be the subgraphs in $\mathcal{H}_{u_0}$ in lex. cyclic order, i.e., labeled such that $(H_r \cap C_L) \subsetneq (H_{r-1} \cap C_L) \subsetneq \ldots \subsetneq (H_1 \cap C_L)$. Since $\mathcal{H}$ is strict-containment free, it follows that $(H_1 \cap C_R) \subsetneq (H_2 \cap C_R) \subsetneq \ldots \subsetneq (H_r \cap C_R)$. Similarly, if $H'_1, \ldots, H'_s$ are the subgraphs in $\mathcal{H}_{v_0}$ in lex. cyclic order, then $(H'_1 \cap C_L) \subsetneq (H'_2 \cap C_L) \subsetneq \ldots \subsetneq (H'_s \cap C_L)$, and $(H'_s \cap C_R) \subsetneq (H'_{s-1} \subsetneq C_R) \ldots \subsetneq (H'_1 \cap C_R)$. Fig. 3.4 shows the subgraphs in $\mathcal{H}_{u_0}$ and in $\mathcal{H}_{v_0}$. We let $\mathcal{H}_L = \{H \in \mathcal{H} : H \subseteq \mathrm{arc}(v_0, u_0)\} \cup \{H_1 \cap C_L\} \cup \{H'_s \cap C_L\}$, and $\mathcal{H}_R = \{H \cap C_R : H \in \mathcal{H}\}$. See Fig. 3.5.

Note that since $\mathcal{H}$ is strict-containment free, so are $\mathcal{H}_L$ and $\mathcal{H}_R$, and that each $H \in \mathcal{H} \setminus \{H_1, H'_s\}$ contributes a run to exactly one of $\mathcal{H}_L$ or $\mathcal{H}_R$, by construction.

By the choice of K_0, in the cycle system $(C_R, \mathcal{H}_R, \mathcal{K}_R)$, each $H \in \mathcal{H}_R$ and each $K \in \mathcal{K}_R$ induces a single run on C_R. Hence, by Lemma 3.2, there is an outerplanar support Q_R for $(C_R, \mathcal{H}_R, \mathcal{K}_R)$ such that $Q_R = C(\mathcal{H}_R)$, the cycle on $\mathcal{H}_R$ in lex. cyclic order.

The cycle system $(C_L, \mathcal{H}_L, \mathcal{K}_L)$ is $axax$-free since $(C, \mathcal{H}, \mathcal{K})$ is $axax$-free. Further, $N(C_L, \mathcal{K}_L) < N$ since d_{K_0} joins two disjoint runs of K_0. Hence, by the inductive hypothesis, $(C_L, \mathcal{H}_L, \mathcal{K}_L)$ admits an outerplanar support Q_L with outer cycle $C(\mathcal{H}_L)$.

Since in Q_L, the outer cycle is $C(\mathcal{H}_L)$ and in Q_R, the outer cycle is $C(\mathcal{H}_R)$, it follows that H_1 and H'_s are consecutive in the outer cycles of Q_L and Q_R. To obtain a support

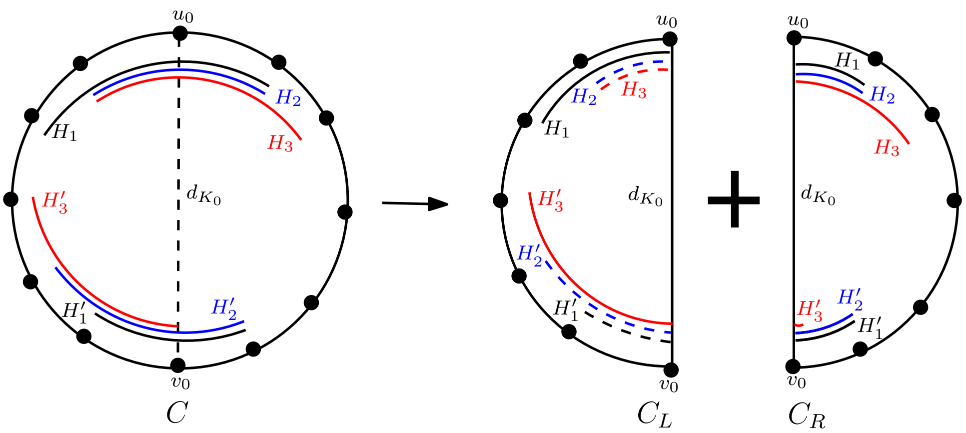
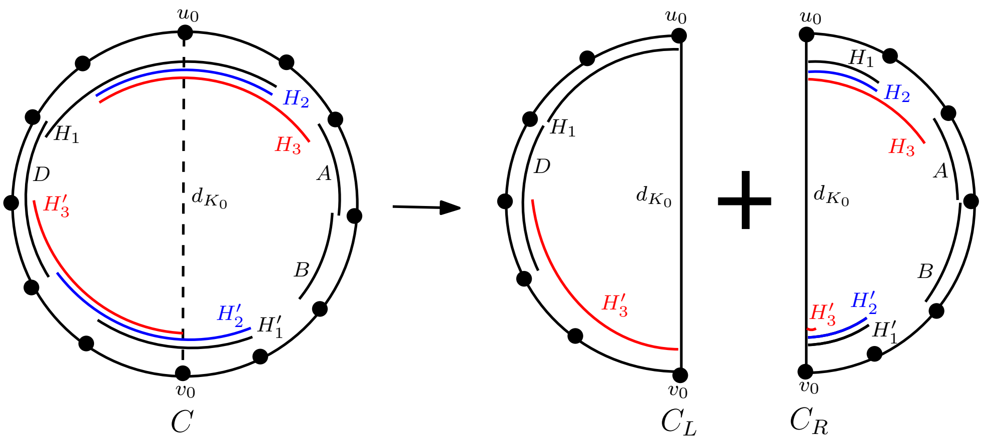


Figure 3.4: Ordering $\mathcal{H}_{u_0}$ and $\mathcal{H}_{v_0}$ subgraphs in C_L and C_R. $(H_3 \cap C_L) \subseteq (H_2 \cap C_L) \subseteq (H_1 \cap C_L)$, and $(H_1 \cap C_R) \subseteq (H_2 \cap C_R) \subseteq (H_3 \cap C_R)$. Analogously, $(H'_1 \cap C_L) \subseteq (H'_2 \cap C_L) \subseteq (H'_3 \cap C_L)$, and $(H'_3 \cap C_R) \subseteq (H'_2 \cap C_R) \subseteq (H'_1 \cap C_R)$.



$(\mathcal{H}, \prec_{lex}) = H_1, H_2, H_3, A, B, H'_1, H'_2, H'_3, D$ $(\mathcal{H}_L, \prec_{lex}) = H_1, H'_3, D$ $(\mathcal{H}_R, \prec_{lex}) = H_1, H_2, H_3, A, B, H'_1, H'_2, H'_3$

Figure 3.5: Construction of $\mathcal{H}_L$ and $\mathcal{H}_R$. Any $H \in \mathcal{H} \setminus \{H_1, H'_3\}$ containing u_0 or v_0 such that $(H \cap C_L) \subseteq (H_1 \cap C_L)$ or $(H \cap C_L) \subseteq (H'_3 \cap C_L)$ is not included in $\mathcal{H}_L$.

Q for $(C, \mathcal{H}, \mathcal{K})$, we identify the copy of H_1 in Q_L and Q_R, and similarly, we identify the copy of H'_s in Q_L and Q_R. It follows that Q is outerplanar and $C(\mathcal{H})$ is the outer cycle in the resulting embedding of Q.

Next, we show that Q is a support. Consider an arbitrary subgraph $K \in \mathcal{K}$. We consider three possible cases for the runs of K.

Suppose the runs of K lie entirely in C_R. Then, $\mathcal{H}_K$ induces a connected subgraph in Q since every subgraph in $\mathcal{H}$ intersecting C_R has a run in $\mathcal{H}_R$, and that Q_R is a support for $(C_R, \mathcal{H}_R, \mathcal{K}_R)$.

Now, suppose the runs of K lie entirely in C_L. If $\mathcal{H}_K \cap (\mathcal{H}_{u_0} \cup \mathcal{H}_{v_0}) = \emptyset$, i.e., if none of the subgraphs in $\mathcal{H}_K$ contains u_0 or v_0, then for each subgraph $H \in \mathcal{H}_K$, we have $H \subseteq \mathrm{arc}(v_0, u_0)$ and thus $H \in \mathcal{H}_L$ since each $H \in \mathcal{H}$ induces a single run on C. Since Q_L is a support for $(C_L, \mathcal{H}_L, \mathcal{K}_L)$, it follows that $\mathcal{H}_K$ induces a connected subgraph of

Q. Now, suppose $\mathcal{H}_K \cap \mathcal{H}_{u_0} \neq \emptyset$, or $\mathcal{H}_K \cap \mathcal{H}_{v_0} \neq \emptyset$. Assume the former, without loss of generality. Since the subgraphs in $\mathcal{H}$ were assumed to be strict-containment free, K intersects a prefix of the sequence of subgraphs in $\mathcal{H}_{u_0}$ in lex. cyclic order, i.e., a prefix of the subgraphs $(H_1, \ldots, H_r)$, since $(H_i \cap C_L) \supseteq (H_{i+1} \cap C_L)$ for $i = 1, \ldots, r-1$. In particular $H_1 \in \mathcal{H}_K$. Again, as argued, the subgraphs in $\mathcal{H}_R$ are strict-containment free, and hence, they appear consecutively in lex. cyclic order on the outer cycle of Q_R. Since Q_R is a support for $(C_R, \mathcal{H}_R, \mathcal{K}_R)$, and Q_L is a support for $(C_L, \mathcal{H}_L, \mathcal{K}_L)$, with H_1 appearing in both Q_L and Q_R, it follows that $\mathcal{H}_K$ induces a connected subgraph of Q. A similar argument holds when both $\mathcal{H}_K \cap \mathcal{H}_{u_0} \neq \emptyset$ and $\mathcal{H}_K \cap \mathcal{H}_{v_0} \neq \emptyset$.

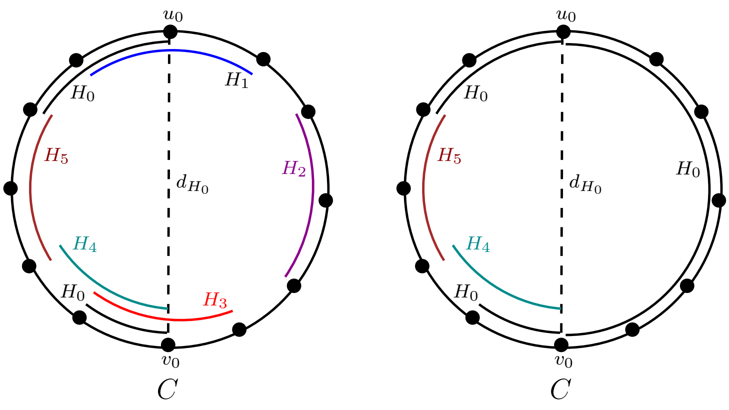
Finally, suppose that K intersects both C_L and C_R. Since $(C, \mathcal{K})$ is $axax$-free and K_0 contains u_0 and v_0, it follows by Proposition 3.3, that K contains either u_0 or v_0. Therefore, either H_1 or H'_s is in $\mathcal{H}_K$. Now, the fact that $\mathcal{H}_K$ induces a connected subgraph in Q is identical to the case when K has runs only in C_L s.t. $\mathcal{H}_K \cap \mathcal{H}_{u_0} \neq \emptyset$. Since K was chosen arbitrarily, Q is a support. ■

Now, we are ready to prove the general case, for which we require a few technical tools. For a cycle system $(C, \mathcal{H}, \mathcal{K})$, let $H_0 = \arg\min_{H \in \mathcal{H}} \ell(H)$. Let $d_{H_0} = \{u_0, v_0\}$ denote the chord of H_0 attaining the minimum. The two *derived cycle systems* $(C, \mathcal{H}_L, \mathcal{K})$ and $(C, \mathcal{H}_R, \mathcal{K})$ corresponding to H_0 are defined as follows:

Let $\mathcal{H}_R = \{H \cap \mathrm{arc}[u_0, v_0] : H \in \mathcal{H} \setminus H_0\} \cup H'_0$, where H'_0 corresponds to H_0, and spans all the vertices of the complementary $\mathrm{arc}[v_0, u_0]$. We construct $\mathcal{H}_L$ in two steps. Set $\mathcal{H}'_L = \{H \cap \mathrm{arc}[v_0, u_0] : H \in \mathcal{H} \setminus H_0\}$. Let $H \in \mathcal{H}$ be s.t. $H \cap \mathrm{arc}(u_0, v_0) \neq \emptyset$. Since $H_0 \cap \mathrm{arc}(u_0, v_0) = \emptyset$ with $u_0, v_0 \in H_0$, and $(C, \mathcal{H})$ is $axax$-free, it follows from Proposition 3.3 that in $\mathrm{arc}[v_0, u_0]$, any run of H must be contained in a run of H_0. Therefore, we remove all such subgraphs from $\mathcal{H}'_L$. We set $\mathcal{H}_L = \mathcal{H}'_L \setminus \{H \in \mathcal{H}'_L : H \cap \mathrm{arc}(u_0, v_0) \neq \emptyset\} \cup H''_0$, where $H''_0 = H_0 \cup \mathrm{arc}[u_0, v_0]$ corresponds to the subgraph H_0. That is, H''_0 extends from u_0 and v_0 to contain the complementary arc $\mathrm{arc}[u_0, v_0]$. This defines the *derived cycle systems* $(C, \mathcal{H}_L, \mathcal{K})$ and $(C, \mathcal{H}_R, \mathcal{K})$ corresponding to the subgraph H_0. Note that each subgraph $H \in \mathcal{H} \setminus \{H_0\}$ has a run in exactly one of $\mathcal{H}_L$ or $\mathcal{H}_R$. Moreover, if $\mathcal{H}$ is containment-free, and $(C, \mathcal{H})$ is $axax$-free, then it follows that $\mathcal{H}_R$ is strict-containment free. Fig. 3.6 shows a partition of $(C, \mathcal{H}, \mathcal{K})$ into the derived systems.

Proposition 3.4. Let $(C, \mathcal{H}, \mathcal{K})$ be a strong $axax$-free cycle intersection system. Let $H_0 = \arg\min_{H \in \mathcal{H}} \ell(H)$ and that $d_{H_0} = \{u_0, v_0\}$, where d_{H_0} is the chord of H_0 attaining the minimum. Then, the two derived cycle systems $(C, \mathcal{H}_L, \mathcal{K})$ and $(C, \mathcal{H}_R, \mathcal{K})$ corresponding to H_0 are also strong $axax$-free.

Proof. Clearly, the cycle systems $(C, \mathcal{H}_L, \mathcal{K})$ and $(C, \mathcal{H}_R, \mathcal{K})$ are $axax$-free. We show that they satisfy the intersection property. By the choice of H_0, in the derived system



(a) $(C, \mathcal{H})$ **(b)** $(C, \mathcal{H}_L)$ **(c)** $(C, \mathcal{H}_R)$

Figure 3.6: Collections $\mathcal{H}_L$ and $\mathcal{H}_R$ of derived systems. $\mathcal{K}$ subgraphs are not shown in the figure.

$(C, \mathcal{H}_R, \mathcal{K})$, each $H \in \mathcal{H}_R$ induces a single run on C. Therefore, it is trivially strong $axax$-free. Consider the derived system $(C, \mathcal{H}_L, \mathcal{K})$. Let $H \in \mathcal{H}_L$ and $K \in \mathcal{K}$ be such that there are vertices h_1, k_1, h_2, k_2 in cyclic order on C with $h_1, h_2 \in H$ and $k_1, k_2 \in K$ such that $H \cap K = \emptyset$. First, suppose that $H = H_0''$. If K has a vertex in $\mathrm{arc}[u_0, v_0]$, then $H \cap K \neq \emptyset$; a contradiction to our assumption. If K does not have a vertex in $\mathrm{arc}[u_0, v_0]$, then it follows that H_0 and K does not satisfy the intersection property in C since H_0 and H_0'' coincide on $\mathrm{arc}[v_0, u_0]$; again a contradiction to the fact that $(C, \mathcal{H}, \mathcal{K})$ is strong $axax$-free.

So, suppose $H \neq H_0''$. By construction of $\mathcal{H}_L$, $H \cap \mathrm{arc}(u_0, v_0) = \emptyset$. This implies H and K do not satisfy the intersection property in $(C, \mathcal{H}, \mathcal{K})$, contradicting the assumption that it is strong $axax$-free. ∎

Now we are done with the necessary tools we require for the proof of the general setting, i.e., when subgraphs in $\mathcal{H} \cup \mathcal{K}$ induce more than one run on C. The proof is similar to that of Lemma 3.3. We use an induction on the total number of runs in the elements of $\mathcal{H}$, like we did for $\mathcal{K}$ in Lemma 3.3. Again, we partition the input cycle into two by choosing a chord of minimum length, but now w.r.t. the family $\mathcal{H}$ instead of $\mathcal{K}$. While the base case will follow directly from Lemma 3.3, we require a combination of Propositions 3.3 and 3.4 to argue the induction step.

Theorem 3.2. Let $(C, \mathcal{H}, \mathcal{K})$ be a strong $axax$-free outerplanar system. Then, there is an outerplanar support Q for $(C, \mathcal{H}, \mathcal{K})$.

Proof. By Corollary 3.1, we assume without loss of generality that $\mathcal{H}$ is containment-free. We prove the result by induction on $N = N(C, \mathcal{H}) = \sum_{H \in \mathcal{H}} (n_H - 1) \geq 0$. If $N = 0$, then each $H \in \mathcal{H}$ induces a single run on C, and by Lemma 3.3, we are done. So, we can assume that $N \geq 1$, i.e., $n_H \geq 2$ for some $H \in \mathcal{H}$. Suppose the theorem holds for all strong $axax$-free systems $(C, \mathcal{H}', \mathcal{K})$ with $N(C, \mathcal{H}') < N$.

Let $(C, \mathcal{H}, \mathcal{K})$ be a strong $axax$-free system with $N(C, \mathcal{H}) = N$. Let $H_0 = \arg\min_{H \in \mathcal{H}} \ell(d_H)$ with $d_{H_0} = \{u_0, v_0\}$, and such that $H_0 \cap \operatorname{arc}(u_0, v_0) = \emptyset$.

Consider the derived cycle systems $(C, \mathcal{H}_L, \mathcal{K})$ and $(C, \mathcal{H}_R, \mathcal{K})$. By Proposition 3.4, both derived cycle systems are strong $axax$-free.

In the cycle system $(C, \mathcal{H}_R, \mathcal{K})$, each subgraph in $\mathcal{H}_R$ induces a single run and $\mathcal{H}_R$ is strict-containment free. By Lemma 3.3, there is an outerplanar support Q_R whose outer cycle is $C(\mathcal{H}_R)$. By the choice of H_0, $N(C, \mathcal{H}_L) < N$, and by Proposition 3.4, $(C, \mathcal{H}_L, \mathcal{K})$ is a strong $axax$-free system. By the inductive hypothesis therefore, $(C, \mathcal{H}_L, \mathcal{K})$ admits an outerplanar support Q_L. By construction of $\mathcal{H}_L$ and $\mathcal{H}_R$, each $H \in \mathcal{H} \setminus \{H_0\}$ has its representative vertices in exactly one of Q_L or Q_R. Let Q be the graph obtained by identifying H_0'' in Q_L and H_0' in Q_R. The graph Q is clearly outerplanar. It remains to show that Q is an intersection support for $(C, \mathcal{H}, \mathcal{K})$.

Let $K \in \mathcal{K}$ be arbitrary. We show that $\mathcal{H}_K$ induces a connected subgraph of Q. We consider three cases for the runs of K. First, suppose that all runs of K lie in $\operatorname{arc}[u_0, v_0]$. Then all subgraphs in $\mathcal{H}$ intersecting K have a representative in $\mathcal{H}_R$. The fact that $\mathcal{H}_K$ is connected in Q follows from the fact that Q_R is an intersection support for $(C, \mathcal{H}_R, \mathcal{K})$.

Now suppose that all runs of K lie entirely in $\operatorname{arc}[v_0, u_0]$. If each $H \in \mathcal{H}_K$ has its representative in $\mathcal{H}_L$, then the fact that $\mathcal{H}_K$ induces a connected subgraph of Q follows from the fact that Q_L is an intersection support for $(C, \mathcal{H}_L, \mathcal{K})$. Otherwise, if an $H \in \mathcal{H}_K$ does not have its representative in $\mathcal{H}_L$, then by construction of $\mathcal{H}_L$, $H \cap \operatorname{arc}(u_0, v_0) \neq \emptyset$. By Proposition 3.3 therefore, H contains u_0 or v_0 and each run of H in $\operatorname{arc}[v_0, u_0]$ is contained in a run of H_0. It follows that $K \cap H_0 \neq \emptyset$ and thus $K \cap H_0'' \neq \emptyset$. Recall that the subgraph in $\mathcal{H}$ containing u_0 or those containing v_0 appear consecutively in the outer cycle $C(\mathcal{H}_R)$ of Q_R, and K intersects a prefix of these sequences of subgraphs. Since K intersects H_0 which contains both u_0 and v_0, $\mathcal{H}_K$ induces a connected subgraph of Q.

Finally, let K intersect both $\operatorname{arc}[u_0, v_0]$ and $\operatorname{arc}[v_0, u_0]$. Note that in this case K intersects H_0 (and hence with H_0' and H_0'') since $u_0, v_0 \in H_0$ and $(C, \mathcal{H}, \mathcal{K})$ is strong $axax$-free. Each $H \in \mathcal{H} \setminus \{H_0\}$ has a representative vertex in one of Q_L or Q_R which are supports for the derived cycle systems with H_0 in both Q_L and Q_R. It follows that $\mathcal{H}_K$ induces a connected subgraph of Q. Since K was chosen arbitrarily, Q is a support. ■

§ 3.4 VC-Dimension and Implementation

In this section, we show a polynomial running time of our algorithm to construct an outerplanar support. We start with the definition of *Vapnik-Chervonenkis dimension*, also called *VC-dimension* for short, a notion introduced by Vapnik and Chervonenkis in 1968. The VC-dimension of a hypergraph is the size of the largest subset of vertices that can be *shattered*-meaning, every possible subset of it can be realized as the intersection with some hyperedge. Mathematically,

Definition 3.5 (VC-dimension)**.** Given a hypergraph $\mathcal{F} = (V, \mathcal{E})$, a subset $Y \subseteq V$ is said to be *shattered* by $\mathcal{E}$ if for each $X \subseteq Y$, there exists $E_X \in \mathcal{E}$ such that $Y \cap E_X = X$. The *VC-dimension* of $\mathcal{F}$ denoted $VCdim(\mathcal{F}) = \max\{|Y| : Y \subseteq V \text{ is shattered}\}$.

This notion captures the combinatorial complexity of the hypergraph and plays a key role in areas like discrete geometry, learning theory, and extremal combinatorics. In particular, it is used in bounding the number of hyperedges in the hypergraph as stated in the following lemma, proved independently by Sauer and Shelah [Sau72]. The lemma was also independently proved by both Vapnik and Chervonenkis (see Chapter 10 in [Mat02]).

Lemma 3.4 (Sauer-Shelah [Sau72; Mat02])**.** Let $\mathcal{F} = (V, \mathcal{E})$ be a hypergraph on n vertices such that $VCdim(\mathcal{H}) = d$. Then $|\mathcal{E}| \leq O(n^d)$.

In our context, we show that for a non-piercing intersection system $(G, \mathcal{H}, \mathcal{K})$, where G is outerplanar, the quantities $|\mathcal{H}|$ and $|\mathcal{K}|$ are polynomially bounded by showing that the VC-dimension of the underlying hypergraph is constant. Possibly the bound here is not sharp, but it is sufficient for our purposes since we only want to establish a polynomial running time. Before that, we recall Definition 2.7, and redefine it as a cycle $abab$-free system.

Definition 3.6 ($abab$-free)**.** Let $(C, \mathcal{H})$ be a cycle system. $H, H' \in \mathcal{H}$ are an $abab$-pair if there are four distinct vertices a_1, b_1, a_2, b_2 in cyclic order on C such that $a_1, a_2 \in H \setminus H'$ and $b_1, b_2 \in H' \setminus H$. $(C, \mathcal{H})$ is $abab$-free if there are no $abab$-pairs in $\mathcal{H}$.

Note 3.4.1. If $(C, \mathcal{H})$ is $axax$-free, then it is $abab$-free, but *not* conversely.

The above notion of $abab$-free cycle systems is equivalent to $ABAB$-free hypergraphs introduced by Eckerman et al. [AKP20]. For our purpose, we write $abab$-free (rather than $ABAB$-free) in the proof of the following theorem, as it better aligns with our terminology. We use the results for hypergraphs defined by points and pseudodisks [AKP20], and the fact that they admit a planar support [RR20].

Theorem 3.6. Let $(G, \mathcal{H}, \mathcal{K})$ be an outerplanar non-piercing graph system. Then, $|\mathcal{H}|, |\mathcal{K}| = O(n^4)$, where $n = |V(G)|$.

Proof. By Lemma 3.1, the cycle system $(C, \mathcal{H}, \mathcal{K})$ is strong $axax$-free, and this implies that $(C, \mathcal{H})$ and $(C, \mathcal{K})$ are $abab$-free.

Ackerman et al. [AKP20] showed that $abab$-free hypergraphs are exactly those that can be represented by a set of pseudodisks that contain a common point in $\mathbb{R}^2$. Hence, the hypergraphs defined by $(C, \mathcal{H})$ can be represented s.t. the vertices of C are points in the plane, and the hyperedges in $\mathcal{H}$ are pseudodisks each containing the origin in such a way that the points of C in a pseudodisk are exactly the points that are the vertices of the corresponding hyperedge.

We show that the VC-dimension of the set system $(C, \mathcal{H})$ is at most 4. The results of Raman and Ray [RR20] imply that for any set P of points and any set $\mathcal{D}$ of pseudodisks, there is a planar support Q on P, i.e., a planar graph on P s.t. the points in D induce a connected subgraph of Q for each $D \in \mathcal{D}$. If a set of 5 points can be shattered, then for each pair of points, there is a hyperedge containing that pair. But, this implies that the support w.r.t. these 5 points is K_5, contradicting the fact that it is planar.

Hence, the VC-dimension of the set system $(C, \mathcal{H})$ is at most 4, and therefore by Lemma 3.4, $|\mathcal{H}| = O(n^4)$. Similarly, $|\mathcal{K}| = O(n^4)$. ∎

Theorem 3.7. If $(G, \mathcal{H}, \mathcal{K})$ is an outerplanar non-piercing intersection system, then an intersection support can be computed in time $O(n^6)$, where $n = |V(G)|$.

Proof. Let $(C, \mathcal{H}, \mathcal{K})$ be the resulting strong $axax$-free cycle system. Then $|C| = n$, and by Theorem 3.6, $|\mathcal{H}|, |\mathcal{K}| = O(n^4)$. We only need to show that our algorithm in Theorem 3.2 runs in time $O(n^6)$.

If each $X \in \mathcal{H} \cup \mathcal{K}$ induces a single run on C, then by Lemma 3.2, $C(\mathcal{H})$ is the desired support. To construct $C(\mathcal{H})$, we walk along C and at each vertex v, add the subgraphs $H \in \mathcal{H}$ to $C(\mathcal{H})$ s.t. $s(H) = v$. For each vertex, we order the subgraphs in increasing order of $t(H)$, which can be computed by sorting the subgraphs H with $s(H) = v$. Thus, $C(\mathcal{H})$ can be computed in time $O(|C| + |\mathcal{H}| \log |\mathcal{H}|) = O(n^4 \log n)$.

For the case when $\mathcal{H}$ consists of single runs and a subgraph in $\mathcal{K}$ can have multiple runs, finding a chord d of smallest length can be done by storing the subgraphs in $\mathcal{K}$ in a heap ordered by $\ell(K)$. The time taken to partition the problem into two sub-problems is $O(|C| \max\{|\mathcal{H}|, |\mathcal{K}|\})$ as we need to go through each subgraph in $\mathcal{H}$ and $\mathcal{K}$, and split the runs into the two sub-problems. We add at most $|C| - 3$ chords in C since the resulting support is outerplanar. Hence, the total running time in this case is $O(|C|^2| \max\{|\mathcal{H}|, |\mathcal{K}|\}) = O(n^6)$.

In the general case, a subgraph in $\mathcal{H}$ can have multiple runs. To compute an $H \in \mathcal{H}$ minimizing $\ell(H)$, we can store the subgraphs in $\mathcal{H}$ in a heap ordered by $\ell(H)$. We can split the problem into two sub-problems in $O(|C| \max\{|\mathcal{H}|, |\mathcal{K}|\})$ time. Since we add at most $|C| - 3$ chords, the overall running time is bounded above by $O(|C|^2 \max\{|\mathcal{H}|, |\mathcal{K}|\})$. Hence, the overall running time of the algorithm is $O(n^6)$. ■

Outerplanar graphs have treewidth at most 2, and the above result shows that if the subgraphs are non-piercing, then the VC dimension of the set system defined by $(C, \mathcal{H})$ is at most 4. We extend this result in Section 4.5 to graphs of bounded treewidth, and we show that for a non-piercing graph system $(G, \mathcal{H})$ of treewidth t, the VC-dimension of set system defined by $(V(G), \mathcal{H})$ is at most $3t + 3$.

§ 3.5 Primal and Dual Supports: Revisited

The existence of an intersection support also implies a primal and a dual support. However, we emphasize here that for the existence of an outerplanar primal or dual support, some weaker conditions are sufficient. In particular, for a primal support, we require the cycle system $(C, \mathcal{H})$ to be $abab$-free and not necessarily $axax$-free, and for a dual support, we require $(C, \mathcal{H})$ to be $axax$-free. This follows directly from the Lemma 2.1 (we re-worded it here in terms of $\mathcal{H}$).

Lemma 3.5. Let $(C, \mathcal{H})$ be an $abab$-free cycle system. Then, we can add a set D of non-intersecting chords in C such that each $H \in \mathcal{H}$ induces a connected subgraph of $C \cup D$. Further, the set D of non-intersecting chords to add can be computed in time $O(mn^4)$, where $n = |C|$, and $m = |\mathcal{H}|$.

Theorem 3.8. Let $(C, \mathcal{H})$ be an $abab$-free cycle system and $c : V(C) \to \{\mathbf{b}, \boldsymbol{r}\}$ be a 2-coloring of the vertices of C. Then there is a primal support for $(C, \mathcal{H})$ that is an outerplanar graph. Further, such a support can be computed in time $O(n^8)$, where $n = |V(C)|$.

Proof. For each $v \in V(C)$ such that $c(v) = \boldsymbol{r}$, we delete it and make its neighbors adjacent. Let C' be the resulting cycle obtained from C, and $\mathcal{H}' = \{H \cap C' : H \in \mathcal{H}\}$. Then $(C', \mathcal{H}')$ is an $abab$-free cycle system. By Lemma 3.5, there is a set D of non-intersecting chords added to C' such that each subgraph in $\mathcal{H}'$ induces a connected subgraph of $C' \cup D$. Thus $C' \cup D$ is an outerplanar primal support for $(C', \mathcal{H}')$. Note that this is also a primal support for $(C, \mathcal{H})$. The claimed running time follows from Lemma 3.5 since $|\mathcal{H}| = O(n^4)$ by Theorem 3.6. ■

Note that Theorem 3.7 implies a primal support in time $O(n^6)$, but Lemma 3.5 gives a primal support in time $O(n^8)$ since $|\mathcal{H}| = O(n^4)$ by Theorem 3.6. This happens since there is a subtle difference between $axax$-free and $abab$-free cycle systems. In $axax$-free, any chord joining two non-consecutive vertices of a subgraph is a non-blocking chord and thus we only need to search a chord of minimum length. In case of $abab$-free cycle system however, we search for a non-blocking chord which requires some extra comparisons during the algorithm to ensure if it is non-blocking or not.

Similar to the primal setting, one may likewise hope that the $abab$-free condition is sufficient to obtain a dual support. Unfortunately, that is not the case as the following example shows: Let G be the graph on $\{1, 2, \ldots 6\}$ as shown in Fig. 3.7. Let $\mathcal{H}$ be subgraphs $H_1 = \{1, 2, 3\}$, $H_2 = \{3, 4, 5\}$, $H_3 = \{5, 6, 1\}$ and $H_4 = \{2, 4, 6\}$. It can be checked that $\mathcal{H}$ is $abab$-free. The dual support for $(G, \mathcal{H})$ is K_4, which is not outerplanar.

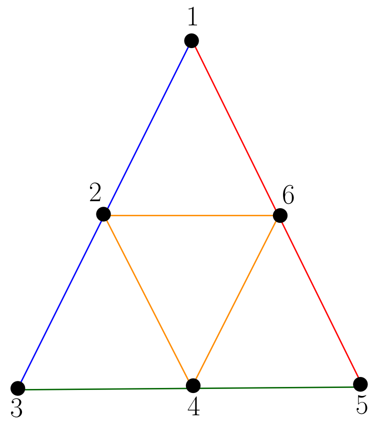


Figure 3.7: $abab$-free cycle system whose only dual support is a complete graph on 4 vertices. The induced subgraphs are shown by four different colors on the edges.

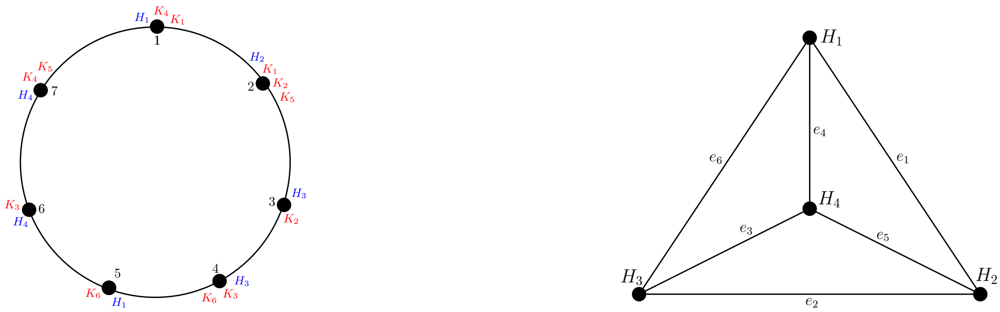
The reason why $(G, \mathcal{H})$ does not admit a dual outerplanar support is that the induced system $(C, \mathcal{H})$ is not $axax$-free - H_4 forms an $axax$-pair with each of H_1, H_2 and H_3. However, $axax$-freeness is a sufficient condition for $(C, \mathcal{H})$ to exhibit an outerplanar dual support, and it follows from Theorem 3.2 since in this case, the condition being $axax$-free coincides with strong $axax$-free.

Theorem 3.9. If $(C, \mathcal{H})$ is $axax$-free, then it admits an outerplanar dual support.

Proof. Let $\mathcal{K} = \{\{v\} : v \in V(C)\}$ be a collection of subgraphs, each consisting of a single vertex. Then $(C, \mathcal{K})$ is $axax$-free. Therefore, the intersection system $(C, \mathcal{H}, \mathcal{K})$ is $axax$-free. Note that the strong $axax$-free property is trivially satisfied by $(C, \mathcal{H}, \mathcal{K})$. By Theorem 3.2, it has an intersection support for Q, which is also a dual support for $(C, \mathcal{H})$ since each $v \in V(C)$ corresponds to a subgraph in $K \in \mathcal{K}$. ∎

Again, we might hope that for an outerplanar intersection system $(G, \mathcal{H}, \mathcal{K})$, if it satisfies the $axax$-free property, then like the primal and dual, we can obtain an outerplanar intersection support. However, the following example shows that in order to construct an intersection support, one needs the strong $axax$-free property: Let $C = (1, 2, \dots 7, 1)$ be a cycle as shown in Fig. 3.8a. Let $\mathcal{H}$ consist of induced subgraphs $H_1 = \{1, 5\}, H_2 = \{2\}, H_3 = \{3, 4\}$ and $H_4 = \{6, 7\}$, and $\mathcal{K}$ consist of induced subgraphs $K_1 = \{1, 2\}, K_2 = \{2, 3\}, K_3 = \{4, 6\}, K_4 = \{1, 7\}, K_5 = \{2, 7\}$ and $K_6 = \{4, 5\}$ of C. It is easy to check that both $(C, \mathcal{H})$ and $(C, \mathcal{K})$ are $axax$-free. However, for each K_i, $i = 1, \dots, 6$, there is exactly one pair H_k, H_ℓ, $k \neq \ell \in \{1, \dots, 4\}$ s.t. H_k and H_ℓ intersect K_i, and hence, the intersection support is a complete graph on 4 vertices (see Fig. 3.8b), which is not outerplanar.

Observe that vertices $1, 4, 5, 6$ are in cyclic sequence with $1, 5 \in H_1$ and $4, 6 \in K_3$ such that $H_1 \cap K_3 = \emptyset$. Similarly, vertices $1, 2, 5, 7$ appear in cyclic sequence with $1, 5 \in H_1$ and $2, 7 \in K_5$ such that $H_1 \cap K_5 = \emptyset$. Hence, the graph system $(C, \mathcal{H}, \mathcal{K})$ does not satisfy the strong $axax$-free property.



(a) Alternate sequences of subgraph H_1 with the subgraphs K_3 and K_5.

(b) Non-outerplanar support. Edge e_i corresponds to K_i for $i = 1 \dots 6$.

Figure 3.8: An example of an $axax$-free, but not strong $axax$-free system $(C, \mathcal{H}, \mathcal{K})$ that does not admit an outerplanar support.

§ 3.6 Concluding Remarks and Open Questions

In this chapter, we studied the existence of an outerplanar support for the primal, dual and intersection hypergraphs that arise from subgraphs of an outerplanar graph. Table 3.1 below summarizes the results proved in this chapter.

It was shown by Buchin et al. [Buc+11] that the decision problem of the existence of 2-outerplanar support is NP-hard, and they left the problem unresolved for the outerplanar case. We proved that for hypergraphs arising from non-piercing subgraphs of an outerplanar graph, there are primal, dual, and intersection supports that are outerplanar. Further, the supports can be computed in time poly(n), where n is the number

	$abab$-free	$axax$-free	Strong $axax$-free
Primal	Yes	Yes	Yes
Dual	No	Yes	Yes
Intersection	No	No	Yes

Table 3.1: The table above summarizes the sufficient conditions required to construct outerplanar support for cycle intersection systems.

of vertices of the host graph. We also weakened the conditions for the existence of primal and dual supports. In particular, by Theorem 3.8, if an abstract hypergraph admits a cyclic ordering of its vertices that is $abab$-free, then it admits an outerplanar support.

The results represented in this chapter are a step forward in addressing the decision problem posed by Buchin et al. However, there are hypergraphs that do not have $abab$-free ordering of their vertices, but they admit an outerplanar support. Indeed, for any $n \geq 4$, consider the complete graph K_n as a hypergraph. If K'_n is a hypergraph obtained by adding a new vertex x to every hyperedge in K_n, then a star with center x is an outerplanar support for K'_n. But K'_n (and also K_n) does not posses $abab$-free ordering.

▷ Limitations.

The scope of the contributions in this chapter is constrained by a few limitations, which we outline here for clarity. First, we could not fully address the unresolved case of Buchin et al. [Buc+11] about the complexity of the existence of an outerplanar support in the abstract setting. Nevertheless, we believe that our approach offers structural insights that could be instrumental in addressing this open case. Second, we don't know how to generalize our results in particular Theorems 3.2 and 3.3 to k-outerplanar graphs. The following open questions may be of interest for further investigation.

> **Open Problem 3.** What are the necessary and sufficient conditions for the existence of an outerplanar support for $\mathcal{H}$?

> **Open Problem 4.** What are the sufficient conditions for the existence of a support that is a 2-outerplanar graph, or more generally, a k-outerplanar graph?

Open Problem 5. If $(G, \mathcal{H}, \mathcal{K})$ is an outerplanar non-piercing system, there is an intersection support that is outerplanar (Theorem 3.3). If G is k-outerplanar, does the non-piercing system $(G, \mathcal{H}, \mathcal{K})$ admit an intersection support that is $c(k)$-outerplanar, where $c(k)$ is a constant that depends only on k?

CHAPTER 4

Support of Bounded Treewidth

Abstract

Let G be a graph of treewidth t and $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ be any two coloring of the vertices of G. We show that if $\mathcal{H}, \mathcal{K}$ are two collections of *non-piercing* subgraphs of G, then the graph system $(G, \mathcal{H})$ admits a primal and a dual support of treewidth respectively $O(2^t)$ and $O(2^{4t})$, and the intersection system $(G, \mathcal{H}, \mathcal{K})$ admits an intersection support of treewidth $2^{O(2^t)}$. The construction is FPT in the treewidth of G. This chapter builds upon the results reported in the following paper.

"Supports for Outerplanar and Bounded Treewidth Graphs".

TABLE OF CONTENTS

§ 4.1 Preliminaries and Notations

Treewidth is a well-known graph parameter introduced by Robertson and Seymour [RS86], during their 20 years of research in proving the *graph minor theorem*. We defined the notion of *tree decomposition* and treewidth in Chapter 1, but for completeness, we include these below.

Definition 4.1 (Tree decomposition)**.** Given a graph $G = (V, E)$, a tree decomposition of G is a pair $(T, \mathcal{B})$, where T is a tree and $\mathcal{B}$ is a collection of *bags* - subgraphs of G indexed by the nodes of T, that satisfies the following properties:

1. For each $v \in V(G)$, the set of bags of T containing v induces a sub-tree of T.
2. For every edge $\{u, v\}$ in G, there is a bag $B \in \mathcal{B}$ such that $u, v \in B$.

Definition 4.2 (Treewidth)**.** The width of a tree decomposition $(T, \mathcal{B})$ is defined to be $\max_{x \in V(T)} |B_x| - 1$. The treewidth of a graph G is the minimum width over all the tree decompositions of G, and is denoted $\text{TW}(G)$.

Throughout this chapter, we use the term *node* to refer to the elements of $V(T)$ for a tree T, and we use *vertices* to refer to the elements of $V(G)$ for a graph G. We use letters x, y, z for the nodes of T, and u, v, w for the vertices of G.

We use two well-known properties of tree decompositions. First, we can assume that T is a rooted binary tree. That is, we can modify a tree decomposition $(T, \mathcal{B})$ to a tree decomposition $(T', \mathcal{B}')$ without increasing the width, where T' is a binary tree. If T is rooted, then we use T_x to denote the subtree of T rooted at node x. The second is that for any edge $e = \{x, y\}$ of T, the set $B_x \cap B_y$ is a *separator* in G. That is, the induced subgraph $G' = G \setminus \{B_x \cap B_y\}$ of G is disconnected. The set $B_x \cap B_y$ is called the *adhesion set* corresponding to edge $\{x, y\}$ of T. It follows from the definition of a tree decomposition that for any clique C in G, there is a bag $B \in \mathcal{B}$ containing all vertices of C. We refer the reader to Chapter 12 in [Die05] for more details on the properties of treewidth and tree decompositions.

For a graph system $(G, \mathcal{H})$, let $(T, \mathcal{B})$ denote a tree decomposition of G of width t. Let $CC(G)$ denote the *chordal completion* of G, i.e., we add edges so that all vertices in a bag are pairwise adjacent. Clearly, a chordal completion does not increase the width of the tree decomposition. Thus if $(T, \mathcal{B})$ is of width $\text{TW}(G)$, then $\text{TW}(CC(G)) = \text{TW}(G)$. Moreover, if $\mathcal{H}$ is a collection of non-piercing subgraphs of G, they remain non-piercing when induced on $CC(G)$ since the underlying hypergraph does not change. Further, it is straightforward that computing a support w.r.t. $CC(G)$ is equivalent to computing a support w.r.t. G. Hence, we assume throughout this chapter that G is a *chordal graph*[1].

We use the following notations: We assume throughout this section that T be a binary tree rooted at node ρ. For any node x of T, we use T_x to denote the sub-tree

[1] A graph is chordal, if there is no induced cycle of length at least 4.

rooted at x. Let $\mathcal{B}_x$ denote the set of bags at the nodes in T_x. The subgraph $G_x = \cup_{z \in T_x} B_z$ of G is the graph induced on the vertices corresponding to the bags in $\mathcal{B}_x$. Note that $(T_x, \mathcal{B}_x)$ induces a tree decomposition of G_x. We use G_{-x} to denote the subgraph induced on $V(G) \setminus V(G_x)$.

Let $\mathcal{H}$ be a collection of subgraphs of G. For $H \in \mathcal{H}$, we let $H|_x = G_x \cap H$ and $\mathcal{H}|_x = \{H|_x : H \in \mathcal{H}\}$. Similarly, $H|_{-x} = H \cap G_{-x}$ and $\mathcal{H}|_{-x} = \{H|_{-x} : H \in \mathcal{H}\}$. Then, $(G_x, \mathcal{H}|_x)$ and $(G_{-x}, \mathcal{H}|_{-x})$ denote the two induced graph systems on G_x and G_{-x} respectively.

For an edge $\{x, y\}$ of T, we use A_{xy} to denote the adhesion set $B_x \cap B_y$. For $A \subseteq V(G)$, let $\mathcal{H}_A = \{H \in \mathcal{H} : A \cap H \neq \emptyset\}$, and for $S \subseteq A$, $\mathcal{H}_S^{=} = \{H \in \mathcal{H}_A : H \cap A = S\}$.

The next two lemmas follow from the fact that the subgraphs considered are non-piercing. For two sets A and B on the same ground set, we say that A and B *properly intersect* if $A \cap B \neq \emptyset$, $A \setminus B \neq \emptyset$ and $B \setminus A \neq \emptyset$.

Lemma 4.1. Let $(G, \mathcal{H})$ be a non-piercing graph system with a tree decomposition $(T, \mathcal{B})$ of G. Let $e = \{x, y\} \in E(T)$ be an edge in T where x is a child of y. Then for $H, H' \in \mathcal{H}_{A_{xy}}$, the following holds: (i) If $A_{xy} \cap H = A_{xy} \cap H'$ and $H|_x \subsetneq H'|_x$, then $H'|_{-x} \subseteq H|_{-x}$. (ii) If $A_{xy} \cap H = A_{xy} \cap H'$, and $H|_x$ and $H'|_x$ properly intersect, then $H|_{-x} = H'|_{-x}$, and (iii) If $H \cap A_{xy} \subsetneq H' \cap A_{xy}$, and $H|_x$ and $H'|_x$ properly intersect, then $H|_{-x} \subseteq H'|_{-x}$.

Proof. The proofs follow immediately from the fact that $\mathcal{H}$ are non-piercing and the adhesion set corresponding to any edge in $E(T)$ is a separator in G. ∎

Lemma 4.2. Let $(G, \mathcal{H})$ be a non-piercing graph system with a tree decomposition $(T, \mathcal{B})$ of G. For any node x of G, the graph system $(G_x, \mathcal{H}|_x)$ is non-piercing and $(T_x, \mathcal{B}_x)$ is a tree decomposition of G_x.

Proof. Since G is chordal, each bag induces a complete subgraph of G. Hence, each $H|_x \in \mathcal{H}|_x$ is connected. If x is not a root of T, let y be the parent of x. Then, for $H|_x, H'|_x \in \mathcal{H}|_x$, the subgraph $H|_x \setminus H'|_x$ of G_x is connected since $H \setminus H'$ is connected and A_{xy} is a separator in G. Therefore, $(G_x, \mathcal{H}|_x)$ is a non-piercing graph system. Finally, that $(T_x, \mathcal{B}_x)$ is a tree decomposition is straightforward. ∎

Recall Lemma 2.5. It immediately implies the following corollary since the class of graphs of treewidth t is closed under adding degree-one vertices.

Corollary 4.1. Let $(G, \mathcal{H}, \mathcal{K})$ be an intersection system. In order to construct a dual support for $(G, \mathcal{H})$ or an intersection support for $(G, \mathcal{H}, \mathcal{K})$ that has treewidth at most t, it is sufficient to construct such a support when $\mathcal{H}$ is containment-free.

Now, we are ready to construct supports for non-piercing graph systems. As in the previous chapters, we construct a primal support first and then a dual support. We use the tools from primal and dual supports to construct an intersection support. As before, the running example from Fig. 1.5 in combination with Remark 2.3.1 serves as a guide to keep in mind different notions of support.

§ 4.2 Primal Support

In this section, we show that a bounded treewidth non-piercing system $(G, \mathcal{H})$ with $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ admits a primal support Q s.t. $\text{TW}(Q) \leq 2^{\text{TW}(G)+2} + \text{TW}(G)$. The proof is algorithmic and yields a polynomial time algorithm if $\text{TW}(G)$ is bounded, i.e., an FPT-algorithm parameterized by $\text{TW}(G)$.

An *easy tree decomposition* of G is a tree decomposition s.t. for each adhesion set A of the tree decomposition and each subgraph H intersecting A, it does so at a blue vertex, i.e., for any adhesion set A, if $H \cap A \neq \emptyset$, then $H \cap \mathbf{b}(A) \neq \emptyset$. If G has an easy tree decomposition of treewidth t, then it is straightforward to construct a primal support Q s.t. $\text{TW}(Q) \leq t$. Moreover, in this case, we only require the subgraphs in $\mathcal{H}$ to be connected.

Lemma 4.3. Let $(G, \mathcal{H})$ be a graph system with $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ such that each $H \in \mathcal{H}$ is a connected subgraph of G. If there is an easy tree decomposition $(T, \mathcal{B})$ of width t, then there is a primal support Q on $\mathbf{b}(V)$ of treewidth at most t.

Proof. Let $(T, \mathcal{B}')$ be the tree decomposition on $\mathbf{b}(V)$ derived from $(T, \mathcal{B})$, where $B' = B \cap \mathbf{b}(B)$ is the bag in $\mathcal{B}'$ corresponding to the bag $B \in \mathcal{B}$. For each $B' \in \mathcal{B}'$, the vertices of B' induce a clique since we work with the chordal completion of G. Then, $(T, \mathcal{B}')$ is the tree decomposition of a graph Q.

To show that Q is a support, consider an $H \in \mathcal{H}$. Since H is a connected subgraph in G, the vertices in $V(H)$ lie in bags corresponding to a connected sub-tree of T. Further, as $(T, \mathcal{B})$ is an easy tree decomposition, $\mathbf{b}(V(H))$ lie in bags of $\mathcal{B}'$ corresponding to a connected sub-tree of T. Since each $B' \in \mathcal{B}'$ induces a complete subgraph of Q, it implies $V(H)$ induces a connected subgraph of Q. ■

In the proof below, we use the following notation: Let ρ be the parent of x in the tree decomposition $(T, \mathcal{B})$. For an adhesion set $A_{x\rho}$ and $S \subseteq A_{x\rho}$, for notational convenience, we assume that each $H \in \mathcal{H}_S^=$ is s.t. $H|_x \cap \mathbf{b}(G_x) \neq \emptyset$ and $H|_{-x} \cap \mathbf{b}(G_{-x}) \neq \emptyset$.

Let $\mathcal{M}_S \subseteq \mathcal{H}_S^{=}$ denote the set of subgraphs H such that $H|_x$ is minimal in G_x in the containment order $\preceq$, i.e., for $H, H' \in \mathcal{H}_S^{=}$, $H|_x \preceq H'|_x \Leftrightarrow H|_x \subseteq H'|_x$. We use $(G_x, \preceq)$ to denote this containment order. We use minimal elements in $\mathcal{M}_S$ to construct an easy tree decomposition at the cost of an increase in the width of the tree decomposition.

Lemma 4.4. Let $(G, \mathcal{H})$ be a non-piercing graph system with $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. A tree decomposition $(T, \mathcal{B})$ of G of width t can be transformed into an easy tree decomposition $(T, \mathcal{B}')$ of width at most $2^{t+2} + t$.

Proof. If $(T, \mathcal{B})$ is an easy tree decomposition, we are done. Otherwise, we modify $(T, \mathcal{B})$ to an easy tree decomposition $(T, \mathcal{B}')$ by adding additional blue vertices to the bags in $\mathcal{B}$. We assume without loss of generality that $(T, \mathcal{B})$ is a binary tree rooted at a node ρ.

We prove by induction on the height of T. If T has height 0, then T consists of a single node ρ and hence for each $H \in \mathcal{H}$, $H \cap \mathbf{b}(B_\rho) \neq \emptyset$. Otherwise, let x and y be children of ρ. By Lemma 4.2, the graph system $(G_x, \mathcal{H}|_x)$ is non-piercing and $(T_x, \mathcal{B}_x)$ is a tree decompositions of G_x. By the inductive hypothesis, there is an easy tree decomposition $(T_x, \mathcal{B}'_x)$ of width at most $2^{t+2} + t$. Analogously, there is an easy tree decomposition $(T, \mathcal{B}'_y)$ of width at most $2^{t+2} + t$ for $(G_y, \mathcal{H}|_y)$.

For each $S \subseteq \boldsymbol{r}(A_{x\rho})$, let $H \in \mathcal{M}_S$. Since $H \cap B'_x \neq \emptyset$ and $(T, \mathcal{B}'_x)$ is an easy tree decomposition, $H \cap \mathbf{b}(B'_x) \neq \emptyset$. Choose a $b \in H \cap \mathbf{b}(B'_x)$ and add it to B_ρ. The tree decomposition remains a valid tree decomposition as the bags containing b correspond to a connected subset of nodes of T. Similarly, for each $S \subseteq \boldsymbol{r}(A_{y\rho})$, choose an $H' \in \mathcal{M}_S$ and a vertex $b' \in H' \cap \mathbf{b}(B'_y)$ and add b' to B_ρ. Let B'_ρ denote the bag at ρ after all subsets of $A_{x\rho}$ and $A_{y\rho}$ have been processed. Since $|A_{x\rho}|, |A_{y\rho}| \leq t+1$ and $|B_\rho| \leq t+1$, we have $|B'_\rho| \leq 2 \cdot 2^{t+1} + (t+1)$. Therefore, width of the tree decomposition $(T, \mathcal{B}')$ is at most $2^{t+2} + t$.

We claim that $(T, \mathcal{B}')$ is an easy tree decomposition. Suppose not. Let $H \in \mathcal{H}$ s.t. $H \cap B_\rho \neq \emptyset$ and $H \cap \mathbf{b}(B'_\rho) = \emptyset$. Then, $H \cap A_{x\rho} \neq \emptyset \neq H \cap A_{y\rho}$ by assumption. Let $H \cap A_{x\rho} = S$ and $H \cap A_{y\rho} = S'$. Let H' be the minimal subgraph in $\mathcal{M}_S$ whose blue vertex was added to B'_ρ and let H'' be the minimal subgraph in $\mathcal{M}_{S'}$ whose blue vertex was added to B'_ρ. Then, H and H' are incomparable in $(G_x, \preceq)$. By part (i) of Lemma 4.1, this implies H and H' are identical in G_{-x}. Similarly, H and H'' are incomparable in $(G_y, \preceq)$ and by Lemma 4.1 H and H'' are identical in G_{-y}. It follows that $H'|_x$ and $H''|_x$ are incomparable in G_x, and $H'|_y$ and $H''|_y$ are incomparable in G_y. Also, note that $H'' \cap A_{x\rho} = S = H' \cap A_{x\rho}$. But this implies $H' \setminus H''$ has at least two connected components - one in G_x and one in G_y, contradicting the assumption that the subgraphs in $\mathcal{H}$ are non-piercing. Hence, $(T, \mathcal{B}')$ is an easy tree decomposition. ∎

Theorem 4.2. Let $(G, \mathcal{H})$ be a non-piercing graph system of treewidth t with $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. There is a primal support Q of treewidth at most $2^{t+2} + t$.

Proof. Let $(T, \mathcal{B})$ be a tree decomposition of G of width t, where T is a binary rooted tree. If $(T, \mathcal{B})$ is an easy tree decomposition, we are done by Lemma 4.3. Otherwise, we transform it into an easy tree decomposition of width at most $2^{t+2} + t$ by Lemma 4.4. We then obtain a primal support Q of the same treewidth by Lemma 4.3. ■

§ 4.3 Dual Support

We now construct a dual support. By Corollary 4.1, we assume that there are *no containments*, i.e., there are no two subgraphs $H, H' \in \mathcal{H}$ such that $H \subseteq H'$. As in the primal setting, we first show how we can obtain a dual support for a simple case. A support for the general case is obtained by reducing it to the simple case. In the following, since we construct a graph on $\mathcal{H}$, we abuse notation and also use H to denote the vertex in the dual support corresponding to $H \in \mathcal{H}$.

For a graph system $(G, \mathcal{H})$, a tree decomposition $(T, \mathcal{B})$ of G is said to be *k-sparse* with respect to $\mathcal{H}$ if for each bag $B \in \mathcal{B}$, at most k subgraphs in $\mathcal{H}$ intersect B. If $(G, \mathcal{H})$ admits such a tree decomposition, we say that $(G, \mathcal{H})$ is k-sparse with respect to $\mathcal{H}$. Note that we do not restrict $\mathcal{H}$ to be non-piercing for $(G, \mathcal{H})$ to be k-sparse.

For a connected graph system $(G, \mathcal{H})$ with a tree decomposition $(T, \mathcal{B})$ of width t that is k-sparse with respect to $\mathcal{H}$, a dual support can be computed in $O(k^2 t |V(G)||\mathcal{H}|)$ time. The algorithm, k**-SDS** that achieves this is as follows: We construct a tree decomposition $(T, \mathcal{B}')$ of a graph Q^*. For each bag $B \in \mathcal{B}$, we put a vertex H in the corresponding bag $B' \in \mathcal{B}'$ if $H \cap B \neq \emptyset$, and then put a complete graph on these vertices. $(T, \mathcal{B}')$ is the desired tree decomposition of Q^* on $\mathcal{H}$. Since each $H \in \mathcal{H}$ is connected and each bag in $\mathcal{B}'$ induces a clique, it is clear that $(T, \mathcal{B}')$ is a valid tree decomposition. Further, since $(T, \mathcal{B})$ is k-sparse, it implies that $(T, \mathcal{B}')$ has treewidth at most k.

Lemma 4.5. For a connected graph system $(G, \mathcal{H})$ and a tree decomposition $(T, \mathcal{B})$ of width t that is k-sparse with respect to $\mathcal{H}$, Algorithm k**-SDS** computes a dual support Q^* with $\textsc{tw}(Q^*) \leq k$.

Proof. For a vertex $v \in V$, consider $\mathcal{H}_v$, the set of all subgraphs containing v. Since $(T, \mathcal{B})$ is a tree decomposition, at each node $x \in V(T)$ s.t. $v \in B_x$, $\mathcal{H}_v \subseteq B'_x$, where B'_x is the bag at node x in $\mathcal{B}'$, i.e., corresponding to B_x. Since the vertices in B'_x are pairwise adjacent, it implies $\mathcal{H}_v$ induces a connected subgraph of Q^*. Hence, Q^* is a dual support. ■

For the general setting, we obtain a dual support by *sparsifying* the input graph system so that it is k-sparse for some k, and such that a support on the sparsified graph system yields a support for the original graph system. The sparsification yields a set $\mathcal{H}'$ of subgraphs that satisfy the following properties: (i) they are in bijective correspondence with $\mathcal{H}$, (ii) each $H' \in \mathcal{H}'$ corresponding to $H \in \mathcal{H}$ is s.t. $H' \subseteq H$. (iii) If $H' \subsetneq H$, then there is an $H'' \in \mathcal{H}$ that *pushed out* H. In this case, $H' = H \setminus H''$ and there is an edge $e = \{u, v\} \in E(G)$ s.t. $u \in H'$ and $v \in H''$. We call e a *connecting edge* between H' and H''.

$\mathcal{H}'$ may contain multiple subgraphs spanning the same set of vertices. We denote by unique$(\mathcal{H}')$ a subset of $\mathcal{H}'$ consisting of one representative from each set of identical subgraphs.

Lemma 4.6. Let $(G, \mathcal{H})$ be a non-piercing graph system, and $(T, \mathcal{B})$ a tree decomposition of G of width t. Then, there is a connected graph system $(G, \mathcal{H}')$ such that $(T, \mathcal{B})$ is a $2^{4(t+1)}$-sparse with respect to unique$(\mathcal{H}')$. If Q' is a dual support for $(G,$ unique$(\mathcal{H}'))$, there is a dual support Q^* for $(G, \mathcal{H}')$ such that $\textsc{tw}(Q^*) = \textsc{tw}(Q')$ and Q^* is also a dual support for $(G, \mathcal{H})$.

Proof. Let $(T, \mathcal{B})$ be a tree decomposition of width t. We assume without loss of generality that T is a binary tree rooted at ρ. To obtain $\mathcal{H}'$, we do a post-order edge traversal of T and at each edge, we do a *pushing*.

The pushing operation is as follows: Let z be the parent of a node x in T. Having done the pushing on the edges in T_x, we do the following at $\{x, z\}$: For each $\emptyset \neq S \subseteq A_{xz}$, choose an $H \in \mathcal{M}_S$. H is called the *pusher* for S at A_{xz}. For each $H' \in \mathcal{H}_S^{=}$ s.t. $H'|_x \setminus H|_x \neq \emptyset$, replace H' by $H'|_x \setminus H|_x$ in $\mathcal{H}$. We say that H' has been *pushed* by H at A_{xz}.

Observe that once a subgraph H' is pushed by a pusher H at A_{xz}, $H'|_x \setminus H|_x \subseteq G_x$, and since pushing is done in a post-order edge traversal of T, H' is pushed at most once. Further, the subgraphs in $\mathcal{H}$ intersecting A_{xz} were non-piercing before pushing and since $H'|_x \setminus H|_x \neq \emptyset$, by part (i) and (ii) of Lemma 4.1, $H'|_{-x} \subseteq H|_{-x}$. Hence, $H'|_x \setminus H|_x = H' \setminus H$. This implies $H'|_x \setminus H|_x$ is a connected subgraph of G. It follows that there is a connecting edge $\{u, v\} \in E(G)$ s.t. $u \in H'|_x \setminus H|_x$ and $v \in H|_x$ as $H'|_x \setminus H|_x \neq \emptyset$ and $H' \cap H \neq \emptyset$.

Let $\mathcal{H}'$ be the subgraphs at the end of the algorithm. Note that by construction $\mathcal{H}'$ is in bijective correspondence with $\mathcal{H}$. For $S \subseteq A_{xz}$, if a subgraph $H' \in \mathcal{H}_S^{=}$ was not pushed by a pusher H at A_{xz}, then $H'|_x = H|_x$. Therefore, the number of distinct subgraphs of G_x in the collection $\mathcal{H}'$ intersecting A_{xz} is less than 2^{t+1}. That is,

$$|\text{unique}(\mathcal{H}'_{A_{xz}}|_x)| < 2^{t+1} \tag{4.1}$$

The subgraphs in unique$(\mathcal{H}')$ intersecting the bag B_x can be associated with 4-tuples based on its intersection with A_{xz}, with the adhesion sets between x and its children, or with the bag B_x itself. From Eq. (4.1), it follows that there are at most $2^{4(t+1)}$ distinct subgraphs in unique$(\mathcal{H}')$ intersecting B_x.

By the arguments above, $(G, \text{unique}(\mathcal{H}'))$ is a connected graph system that is at most $2^{4(t+1)}$-sparse, and hence by Lemma 4.5, it has a dual support Q' of treewidth at most $2^{4(t+1)}$. By Corollary 4.1, we can extend Q' to a support Q^* for $(G, \mathcal{H}')$ without increasing the treewidth.

We now argue that Q^* is a dual support for $(G, \mathcal{H})$. Consider a vertex $v \in V(G)$. The algorithm above ensures that each subgraph is pushed at most once. Suppose $H' \in \mathcal{H}_v$ was pushed by H so that its modified copy $H'' = H' \setminus H$ does not cover v. Then, $H \in \mathcal{H}_v$. Let $e = \{a, b\}$ be the connecting edge between H and H'' such that $a \in H$ and $b \in H''$. Since $(T, \mathcal{B})$ is a valid tree-decomposition, there is a bag B containing both a and b. Let H_1 be the unique representative of H'' in unique$(\mathcal{H}')$. Since Algorithm k-**SDS** puts a complete graph on the subgraphs intersecting B, it implies H and H_1 are adjacent in Q'. In Q^*, we made H' adjacent to H_1. Hence, Q^* is a dual support for $(G, \mathcal{H})$. ■

Now we are ready to prove the main result of this section.

Theorem 4.3. Let $(G, \mathcal{H})$ be a non-piercing graph system of treewidth t. There is a dual support Q^* of treewidth at most $2^{4(t+1)}$.

Proof. Let $(T, \mathcal{B})$ be a tree decomposition of G of width t. If $(T, \mathcal{B})$ is $2^{4(t+1)}$-sparse, then we are done by Lemma 4.5. Otherwise, by Lemma 4.6, we obtain a dual support Q^* for $(G, \mathcal{H})$, of treewidth at most $2^{4(t+1)}$. ■

§ 4.4 Intersection Support

In this section, we obtain an intersection support of treewidth $2^{O(2^{\text{TW}(G)})}$ for a non-piercing intersection system $(G, \mathcal{H}, \mathcal{K})$. The construction of the intersection support uses the construction of both the primal and dual support, and this leads to the double exponential bound on the treewidth of the intersection support.

Let $(T, \mathcal{B})$ be a tree decomposition of G. $(T, \mathcal{B})$ is said to be a $\mathcal{K}$-*easy* tree decomposition if for each $K \in \mathcal{K}$ and each adhesion set A with $A \cap K \neq \emptyset$, there is an $H \in \mathcal{H}_K$

s.t. $H \cap (K \cap A) \neq \emptyset$. Similar to the setting for the dual support, we say that $(T, \mathcal{B})$ is k-sparse with respect to $\mathcal{H}$ if for each bag $B \in \mathcal{B}$, there are at most k subgraphs in $\mathcal{H}$ that intersect B.

We start by showing that if $(T, \mathcal{B})$ is a $\mathcal{K}$-easy k-sparse tree decomposition, then there is an intersection support of treewidth at most k. The proof follows along the same lines as the proofs of Lemmas 4.3 and 4.5.

Lemma 4.7. Let $(G, \mathcal{H}, \mathcal{K})$ be a connected intersection system. If $(T, \mathcal{B})$ is a $\mathcal{K}$-easy tree decomposition of G such that it is k-sparse with respect to $\mathcal{H}$, then there is an intersection support for $(G, \mathcal{H}, \mathcal{K})$ of treewidth at most k.

Proof. We construct a tree decomposition $(T, \mathcal{B}')$ of a graph $\tilde{Q}$ of width at most k. For each bag $B \in \mathcal{B}$ and each subgraph $K \in \mathcal{K}$ intersecting B, we put a vertex in bag $B' \in \mathcal{B}'$ for each $H \in \mathcal{H}_K$ s.t. H intersects B, where B' is the bag in $(T, \mathcal{B}')$ corresponding to the bag B in $(T, \mathcal{B})$. Since $(T, \mathcal{B})$ is k-sparse, there are at most k vertices of $\mathcal{H}$ in B'. We put a complete graph on the vertices of $\mathcal{H}$ in B' and obtain a tree decomposition $(T, \mathcal{B}')$ of a graph $\tilde{Q}$ on $\mathcal{H}$. Since each $H \in \mathcal{H}$ is connected, the vertex in $\tilde{Q}$ corresponding to H, lies in a connected set of bags of $(T, \mathcal{B}')$. Further, by construction, each edge between a pair of vertices H, H' of $\tilde{Q}$ lies in some bag of $\mathcal{B}'$. Hence, $(T, \mathcal{B}')$ is a valid tree decomposition of $\tilde{Q}$.

We now show that $\tilde{Q}$ is an intersection support. Consider any $K \in \mathcal{K}$. Since K is a connected subgraph of G, the vertices of K lie in a sub-tree of T. Since $(T, \mathcal{B})$ is $\mathcal{K}$-easy, for each adhesion set intersected by K, there is a subgraph $H \in \mathcal{H}_K$ intersecting that adhesion set. This implies that $\mathcal{H}_K$ induces a connected subgraph of $\tilde{Q}$, and hence $\tilde{Q}$ is an intersection support for $(G, \mathcal{H}, \mathcal{K})$. ■

We are now ready to prove the main result of this section - the construction of an intersection support. In the proof, we demonstrate how a non-piercing system can be modified so that it satisfies the conditions of Lemma 4.7, i.e., we transform $(G, \mathcal{H}, \mathcal{K})$ to an intersection system admitting a $\mathcal{K}$-easy tree decomposition that is k-sparse for some appropriate choice of k. By leveraging the support for the transformed system, we then obtain the required support for $(G, \mathcal{H}, \mathcal{K})$.

Theorem 4.4. Let $(G, \mathcal{H}, \mathcal{K})$ be a non-piercing intersection system of treewidth t. Then there is an intersection support $\tilde{Q}$ of treewidth at most $2^{2^{t+4}+4(t+1)}$.

Proof. Let $(T, \mathcal{B})$ be a tree decomposition of G of width t. Suppose $(T, \mathcal{B})$ is not $\mathcal{K}$-easy. We define $\phi : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. For each $v \in V(G)$, if $\mathcal{K}_v \neq \emptyset$ and $\mathcal{H}_v \neq \emptyset$, we set $\phi(v) = \mathbf{b}$. Otherwise, set $\phi(v) = \boldsymbol{r}$. Under this coloring, since $(T, \mathcal{B})$ is not $\mathcal{K}$-easy, there is a subgraph $K \in \mathcal{K}$ and an adhesion set A s.t. $K \cap A \subseteq \boldsymbol{r}(A)$. Since the subgraphs in $\mathcal{K}$ are non-piercing, by Lemma 4.4, we obtain a tree decomposition $(T, \mathcal{B}')$ of treewidth $t' = 2^{t+2} + t$ that is easy with respect to $\mathcal{K}$. By the choice of coloring, it follows $(T, \mathcal{B}')$ is $\mathcal{K}$-easy.

Each $K \in \mathcal{K}$ induces a connected subgraph of G. Hence, K intersects a connected set of bags of $(T, \mathcal{B}')$. Since $(T, \mathcal{B}')$ is $\mathcal{K}$-easy, for any pair of vertices $u, v \in K$, there is a path in T between a bag containing u and a bag containing v s.t. each adhesion set on this path is intersected by a subgraph $H \in \mathcal{H}_K$.

Recall that to obtain dual support for $(G, \mathcal{H})$, the Algorithm k-**SDS** adds a complete graph on the subgraphs in $\mathcal{H}$ intersecting each bag of $(T, \mathcal{B}')$. Therefore, a dual support for $(G, \mathcal{H})$ thus obtained is also an intersection support for $(G, \mathcal{H}, \mathcal{K})$. However, this support may not have bounded treewidth. To obtain a support of small treewidth, we need to first sparsify the subgraphs in $\mathcal{H}$.

Since the subgraphs in $\mathcal{H}$ are non-piercing, by Lemma 4.6, we can obtain a collection $\mathcal{H}'$ s.t. $(T, \mathcal{B}')$ is $2^{4(t'+1)}$-sparse with respect to unique$(\mathcal{H}')$. Note that $(T, \mathcal{B}')$ remains $\mathcal{K}$-easy w.r.t. the intersection system $(G, \text{unique}(\mathcal{H}'), \mathcal{K})$. Now, by Algorithm k-**SDS** we obtain a tree decomposition of a graph Q^* that by Lemma 4.5 is a dual support for $(G, \text{unique}(\mathcal{H}'))$ and $\text{TW}(Q^*) \leq 2^{4(t'+1)} = 2^{2^{t+4}+4(t+1)}$. Since Q^* is obtained by k-**SDS**, it also an intersection support for $(G, \text{unique}(\mathcal{H}'), \mathcal{K})$. By Theorem 4.3, we can extend Q^* to a dual support $\tilde{Q}$ for $(G, \mathcal{H})$ such that $\text{TW}(\tilde{Q}) = \text{TW}(Q^*)$. Then $\tilde{Q}$ is also an intersection support for $(G, \mathcal{H}, \mathcal{K})$ since the tree decomposition $(T, \mathcal{B}')$ is $\mathcal{K}$-easy. ■

§ 4.5 Implementation

In this section, we show that our algorithms for the construction of supports of bounded treewidth run in polynomial time if the treewidth of the host graph is bounded. We argue by showing a result below that generalizes Theorem 3.6 to graphs of treewidth t. We show that if the treewidth of G is bounded, then $|\mathcal{H}|$ and $|\mathcal{K}|$ are bounded by poly(n), where $n = |V(G)|$.

Theorem 4.5. Let $(G, \mathcal{H}, \mathcal{K})$ be a non-piercing intersection system of treewidth t. Then $|\mathcal{H}|, |\mathcal{K}| = O(n^{3t+3})$.

Proof. We show that $|\mathcal{H}| = O(n^{3t+3})$. The bound on the size of $\mathcal{K}$ follows analogously.

First, we will show that the VC-dimension of the set system defined by $(V(G), \mathcal{H})$ is at most $3t + 3$. Let $(T, \mathcal{B})$ be a tree decomposition of G of width t. We will show that no set of size more than $3t + 3$ can be shattered. Suppose not. Let S be a set with $3t + 4$ vertices that can be shattered. There is an edge $\{x, y\}$ in T such that the union of bags in T_x contains a subset $S_1 \subsetneq S$ of at least $t+2$ vertices, and the union of bags in $T \setminus T_x$ contains a subset $S_2 \subsetneq S$ of at least $t+2$ vertices, where $S_1 \cap S_2 = \emptyset$. Note that such subsets should exist since each bag contains at most $t+1$ vertices and $|S| = 3t+4$.

There are at least $k = \binom{t+2}{\lfloor \frac{t+2}{2} \rfloor}$ subsets of S_i of size $\lfloor \frac{t+2}{2} \rfloor$ for $i \in \{1, 2\}$. Let $H_1, H_2, \ldots H_k$ be the distinct subgraphs in $\mathcal{H}$ defined by these k sets in S_1, and $H_1', H_2', \ldots H_k'$ be the distinct subgraphs in $\mathcal{H}$ defined by these k sets in S_2. Note that H_i's are pairwise incomparable in the containment order. Similarly, H_i''s are pairwise incomparable.

Next, we define a set of subgraphs in $\mathcal{H}$ that intersect with both S_1 and S_2. Such a set of subgraphs must exist since S is shattered by elements of $\mathcal{H}$. Let $J_i \in \mathcal{H}$ be a subgraph such that $J_i \cap S = H_i \cup H_i'$ for $i = 1, 2, \ldots k$. Then, each J_i contains exactly $2\lfloor \frac{t+2}{2} \rfloor$ vertices of S and hence, they are pairwise incomparable in S. Also, each J_i intersects the adhesion set A_e as they are connected subgraphs. Since the subgraphs in $\mathcal{H}$ are non-piercing, by part (ii) of Lemma 4.1, J_i and J_ℓ should be incomparable in A_e for $i \neq \ell$. But there are only $\binom{t+1}{\lfloor \frac{t+1}{2} \rfloor} < k$ subsets of A_e that are pairwise incomparable. This contradicts the fact that the set S can be shattered. Hence, the VC-dimension of $(V(G), \mathcal{H})$ is at most $3t+3$. By Sauer-Shelah Lemma 3.4 therefore, $|\mathcal{H}| = O(n^{3t+3})$. ■

Now, we are ready to show the running time of our algorithms.

Theorem 4.6. Let G be an n-vertex graph of treewidth t, and $\mathcal{H}, \mathcal{K}$ be non-piercing subgraphs of G. Then, a primal and a dual support of treewidth $O(2^t)$ and $O(2^{4t})$ respectively can be computed in time poly(n^t). And, an intersection support of treewidth at most $2^{O(2^t)}$ can be computed in time poly(n^{2^t}).

Proof. By Theorems 4.2 to 4.4, there are appropriate supports of the claimed treewidth. We show below the running time of the algorithms.

Since $\text{TW}(G) = t$, a tree decomposition $(T, \mathcal{B})$ of width t can be computed in time $2^{O(t)}n$, where T is a binary tree and has $O(n)$ number of nodes [BK96; Bod98; Bod+16].

For the construction of a primal support, we require an easy tree decomposition of G. We did a post-order traversal of the edges in T and chose a minimal subgraph in $\mathcal{M}_S$ for each non-empty subset S of an adhesion set A in T. Since A has at most $t+1$ vertices, there are at most 2^{t+1} choices for S. Also, by Theorem 4.5, $|\mathcal{H}| = O(n^{3t+3})$. Therefore, a primal support of treewidth $2^{t+2} + t$ can be computed in time poly(n^t).

For a dual support, we first construct a graph system that is $2^{4(t+1)}$-sparse. For such a construction, we again do a post-order traversal of the edges in T and choose a subset S of an adhesion set A to select a pusher in $\mathcal{M}_S$. In $O(|\mathcal{H}|^2)$ time, we can do the pushing operation at a subset S to get new subgraphs. Therefore, all the adhesion sets can be processed in time $O(2^t n |\mathcal{H}|^2)$. Once a $2^{4(t+1)}$-sparse tree decomposition is obtained, the algorithm k-**SDS** puts a complete graph on the subgraphs in each bag and hence can be done in $O(k^2 t)$ time, where $k = 2^{4(t+1)}$. Therefore, a dual support of treewidth $2^{4(t+1)}$ can also be computed in time poly(n^t) since $|\mathcal{H}| = O(n^{3t+3})$ by Theorem 4.5.

Finally, the construction of an intersection support follows the construction of primal and dual supports. By the arguments above, it follows that a $\mathcal{K}$-easy tree decomposition can be computed in time poly(n^t) where the width of the resulting tree decomposition is $O(2^t)$. Then a $2^{O(2^t)}$-sparse tree decomposition can be computed in time poly(n^{2^t}). Hence, the total running time to compute an intersection support is poly(n^{2^t}). ∎

§ 4.6 Lower Bounds on Treewidth of Primal and Dual Supports

In this section, we show that there exist non-piercing graph systems for which the treewidth of any primal or dual support is exponential in the treewidth of the host graph.

Theorem 4.7. For any $t \in \mathbb{N}$, there is a non-piercing graph system $(G, \mathcal{H})$ with $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$ such that $\text{TW}(G) \leq t$, but $\text{TW}(Q) \geq \frac{2^{\lfloor t/2 \rfloor}}{\sqrt{t}}$ for any primal support Q.

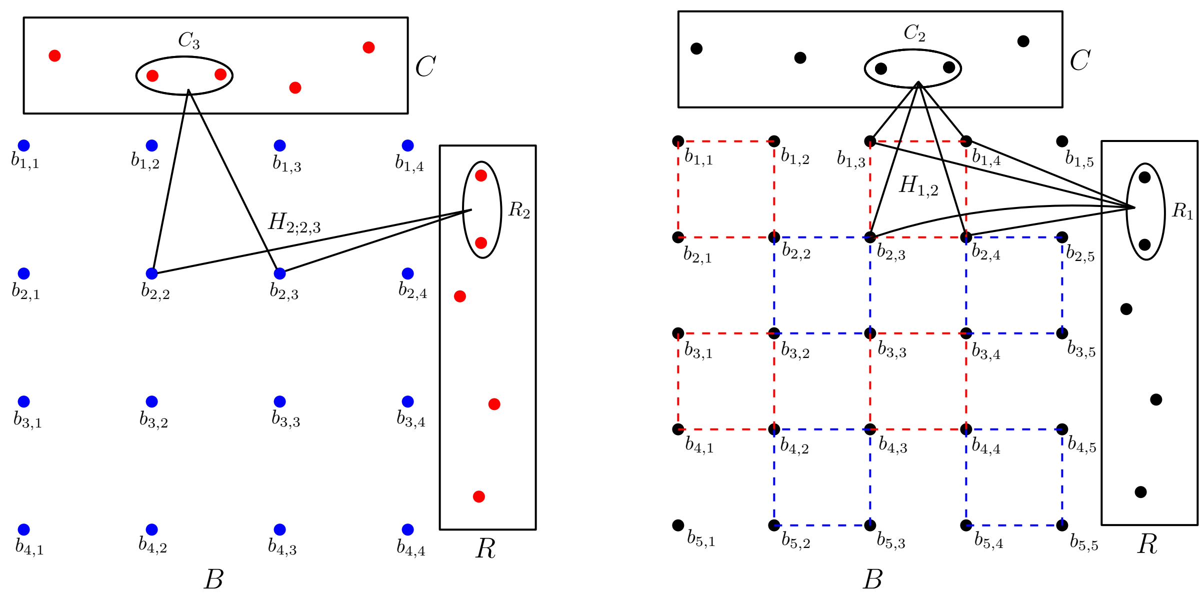
Proof. Let $n = \lfloor t/2 \rfloor$ and let $N = \binom{n}{\lfloor n/2 \rfloor}$. We construct an $N \times N$ grid B of isolated vertices $b_{i,j}$, $i = 1, \ldots, N$ and $j = 1, \ldots, N$. This constitutes the set of vertices colored $\mathbf{b}$. Let $R = \{r_1, \ldots, r_n\}$ and $C = \{c_1, \ldots, c_n\}$ be two sets of n isolated vertices each. The vertices in $R \cup C$ are colored R. Let $\mathcal{R} = \{R_1, \ldots, R_N\}$ be the collection of $\binom{n}{\lfloor n/2 \rfloor}$ subsets of R of size $\lfloor n/2 \rfloor$. Similarly, let $\mathcal{C} = \{C_1, \ldots, C_N\}$ be the $\binom{n}{\lfloor n/2 \rfloor}$ subsets of C of size $\lfloor n/2 \rfloor$. Let the host graph G be the complete bipartite graph with bipartition $(R \cup C)$ and B.

For $i = 1, \ldots, N$ and $j = 1, \ldots, N-1$, let $H_{i;j,j+1}$ be the subgraph induced on the vertices $\{b_{i,j}, b_{i,j+1}\} \cup R_i \cup C_{j+1}$ for $i = 1, \ldots, N$ and $j = 1, \ldots, N-1$. Similarly, for each $i = 1, \ldots, N-1$ and $j = 1, \ldots, N$, let $H_{i,i+1;j}$ be the subgraph induced on the vertices $\{b_{i,j}, b_{i+1,j}\} \cup R_{i+1} \cup C_j$. See Fig. 4.1a. Let $\mathcal{H} = \{H_{i;j,j+1} : i = 1, \ldots, N, j = 1, \ldots, N-1\} \cup \{H_{i,i+1;j} : i = 1, \ldots, N-1, j = 1, \ldots, N\}$.

By construction, each subgraph in $\mathcal{H}$ contains exactly two vertices colored $\mathbf{b}$ corresponding to vertices in the grid that are consecutive row-wise or column-wise, and one subset from each of the collections $\{R_1, \ldots, R_N\}$ and $\{C_1, \ldots, C_N\}$. Since G is a complete bipartite graph, for any $H \in \mathcal{H}$, the two blue vertices in H are adjacent to all red vertices in H. Hence, H is a connected subgraph of G.

Consider two distinct subgraphs $H, H' \in \mathcal{H}$. Since H and H' differ in either their row index or column index, $H \setminus H'$ and $H' \setminus H$ each contains at least one blue vertex. Suppose H and H' differ in their row index. By construction, H and H' are adjacent to distinct subsets of size $\lfloor n/2 \rfloor$ in R. Hence, both $(H \setminus H') \cap R$ and $(H' \setminus H) \cap R$ are non-empty. Similarly, if H and H' differ in their column index, then $(H \setminus H') \cap C$ and $(H' \setminus H) \cap C$ are non-empty. Since $H \setminus H'$ contains at least one blue vertex of H and G is a complete bipartite graph, each red vertex in $H \setminus H'$ is connected to the blue vertices in $H \setminus H'$. On the other hand, if both blue vertices of H are present in $H \setminus H'$, then they are adjacent via a red vertex in $H \setminus H'$ since $H \setminus H' \cap R$ or $H \setminus H' \cap C$ is non-empty and G is a complete bipartite graph. Thus, $H \setminus H'$ is connected. A symmetric argument implies $H' \setminus H$ is connected and hence $\mathcal{H}$ is a set of non-piercing subgraphs of G. Since G is a complete bipartite graph, each blue vertex in $H \setminus H'$ is adjacent to each red vertex in $H \setminus H'$. Thus, $H \setminus H'$ is connected. Similarly, $H' \setminus H$ is connected and hence, $(G, \mathcal{H})$ is a non-piercing graph system.

Since G is a bipartite graph with $2n$ red vertices, G has treewidth at most $2n \leq t$. There is a subgraph in $\mathcal{H}$ corresponding to each pair of blue vertices that are consecutive along a row of the grid or along a column of the grid. Therefore, any primal support contains a grid of size $N \times N$ as a subgraph. Since $N = \binom{n}{\lfloor n/2 \rfloor} \geq \frac{2^n}{\sqrt{2n}}$, it follows that the treewidth of any primal support is at least $\frac{2^n}{\sqrt{2n}} \geq \frac{2^{\lfloor t/2 \rfloor}}{\sqrt{t}}$. ∎



(a) Construction of primal hypergraph **(b)** Construction of dual hypergraph

Figure 4.1: Construction of exp. lower bounds on treewidth for primal and dual supports. (a) Subgraph $H_{2;2,3}$ consists of vertices $\{b_{2,2}, b_{2,3}\} \cup R_2 \cup C_3$. (b) Subgraph $H_{1,2}$ consists of vertices $\{b_{1,3}, b_{2,3}, b_{1,4}, b_{2,4}\} \cup R_1 \cup C_2$.

Theorem 4.8. For any $t \in \mathbb{N}$, there is a non-piercing graph system $(G, \mathcal{H})$ such that $\text{TW}(G) \leq t$ and for any dual support Q^*, $\text{TW}(Q^*) \geq \frac{2^{\lfloor t/2 \rfloor}}{\sqrt{t}}$.

Proof. The construction is similar to that of the primal lower bound. Let $n = \lfloor t/2 \rfloor$ and let $N = \binom{n}{\lfloor n/2 \rfloor}$. Let B be a $(2N+1) \times (2N+1)$ grid of isolated vertices $b_{i,j}$ for $i, j \in \{1, \ldots 2N+1\}$. Let R and C be two sets of n isolated vertices in each. The host graph is the complete bipartite graph G with bipartition $(R \cup C)$ and B. Let $\mathcal{R} = \{R_1, \ldots, R_N\}$ denote the $\binom{n}{\lfloor n/2 \rfloor}$ subsets of R of size $\lfloor n/2 \rfloor$. $\mathcal{C} = \{C_1, \ldots, C_N\}$ is defined analogously.

We define the subgraphs $H_{i,j}$ so that each subgraph contains 4 vertices of B in a checkerboard pattern. More formally, the subgraphs $H_{i,j}$ are defined as follows. Let $i \in \{1, \ldots, 2N\}$. If i is odd, then for $j = 1, \ldots, N$, $H_{i,j}$ is the induced graph on $b_{i,2j-1}, b_{i+1,2j-1}, b_{i,2j}, b_{i+1,2j}$ and the subset R_i of R and C_j of C. If i is even, then for $j = 2, \ldots, N$, $H_{i,j}$ is the induced subgraph of G on the vertices $b_{i,2j}, b_{i+1,2j}, b_{i,2j+1}, b_{i+1,2j+1}$ and the subsets R_i and C_j. See Fig. 4.1b. For odd (even) i, the vertices of $H_{i,j}$ in B are shown by red (blue) squares. Let $\mathcal{H}$ denote the set of subgraphs thus constructed.

Each subgraph $H_{i,j} \in \mathcal{H}$ is induced on four vertices of B and a subset R_i of R and a subset C_j of C. Since G is a complete bipartite graph, it follows that each subgraph $H \in \mathcal{H}$ induces a connected subgraph of G. Consider any two subgraphs $H, H' \in \mathcal{H}$. Since they differ in at least one of the row or column indices, it follows that $H \setminus H'$ and $H' \setminus H$ each contain at least three vertices of B. Further, if H and H' differ in their row index, then they contain distinct subsets in $\mathcal{R}$. Similarly, if H and H' differ in their column index, they contain distinct subsets in $\mathcal{C}$. Since no two subsets in $\mathcal{R}$ are contained in one another, it follows that $(H \setminus H') \cap R$ and $(H' \setminus H) \cap R$ are non-empty if H and H' differ in their row index. Similarly, $(H \setminus H') \cap C$ and $(H' \setminus H) \cap C$ are non-empty if H and H' differ in their column index. Since G is a complete bipartite graph and $H \setminus H'$ contain vertices of B and vertices of either $R \cup C$, it follows that $H \setminus H'$ is connected. Similarly, $H' \setminus H$ is connected. Hence, $\mathcal{H}$ is a collection of non-piercing subgraphs of G.

Since $|R \cup C| = 2n$, it follows that $\text{TW}(G) \leq 2n \leq t$. On the other hand, for $i = 2, \ldots, 2N$ and $j = 2, \ldots, 2N$, the vertices $b_{i,j}$ are contained in exactly two subgraphs in $\mathcal{H}$, and thus, they have to be adjacent in any dual support for $(G, \mathcal{H})$. This implies that any dual support Q^* must contain a grid of size $N \times N$ as an induced subgraph and therefore, $\text{TW}(Q^*) \geq N$. Since $N = \binom{n}{\lfloor n/2 \rfloor}$, $\text{TW}(Q^*) \geq \frac{2^n}{\sqrt{2n}} \geq \frac{2^{\lfloor t/2 \rfloor}}{\sqrt{t}}$. ∎

§ 4.7 Concluding Remarks and Open Questions

In this chapter, we studied support for hypergraphs arising from non-piercing subgraphs of a graph G of bounded treewidth, and generalized the results presented in Chapter 3, namely, the construction of an outerplanar intersection support for a non-piercing outerplanar graph system. We proved that if G has treewidth t, then there are primal and dual supports having treewidth exponential in the parameter t, and an intersection support that has treewidth double exponential in t. All our algorithms run in polynomial time in the number of vertices of G if t is bounded above by a constant. We also construct examples where the exponential blow-up in the treewidth of any primal or dual support is essential.

Table 4.1 below summarizes the results presented in this chapter.

	Upper bound	**Lower bound**
Primal	$O(2^t)$	$\frac{2^{\lfloor t/2 \rfloor}}{\sqrt{t}}$
Dual	$O(2^{4t})$	$\frac{2^{\lfloor t/2 \rfloor}}{\sqrt{t}}$
Intersection	$2^{O(2^t)}$	$\frac{2^{\lfloor t/2 \rfloor}}{\sqrt{t}}$

Table 4.1: The table shows bounds on the treewidth of primal, dual, and intersection supports for non-piercing graph/intersection systems of treewidth t.

▷ Limitations.

While the methods developed in this chapter provide new insights into the topic, there remain some limitations that warrant attention. In particular, to reduce the gap between the upper and lower bounds on the treewidth of both primal and dual supports (see Table 4.1). Another issue is with the double exponent of treewidth of intersection support, and we don't know how close this is to the optimal treewidth of an intersection support for a non-piercing graph system. Several open questions arise from this work, including the following:

> **Open Problem 6.** For non-piercing graph systems $(G, \mathcal{H})$ of treewidth t, there are gaps between the lower and upper bounds on the treewidth of both primal and dual supports as shown in Table 4.1 (see Theorems 4.2, 4.3, 4.7 and 4.8 for reference). This motivates further investigation into whether the current gaps between the bounds can be narrowed or eliminated through improved constructions.

Open Problem 7. We know that if $(G, \mathcal{H}, \mathcal{K})$ is a non-piercing graph system of treewidth t, it admits an intersection support of treewidth $2^{O(2^t)}$ (Theorem 4.4). Is there an intersection support that is a single exponential in t? Or, is there a construction similar to the primal and dual settings shown in Theorems 4.7 and 4.8 that implies a double exponent on the treewidth of an intersection support?

From our results in Chapter 3, it follows that a non-piercing outerplanar graph system admits an outerplanar support. Since $\text{TW}(G) = 2$ for any outerplanar graph G, we wonder if for a non-piercing graph system of treewidth 2, there exists a primal/dual or intersection support of treewidth at most 2.

Open Problem 8. Outerplanar graphs have treewidth at most 2. We saw in Chapter 3 that if G is outerplanar and $\mathcal{H}, \mathcal{K}$ are non-piercing, then the primal, dual and intersection hypergraphs admit outerplanar supports. However, we proved in Theorems 4.7 and 4.8 that for a non-piercing graph system of treewidth t, a primal and dual support can have treewidth exponential in t. Is there a 2-outerplanar support if G has treewidth 2? Does the result hold if G is a *series parallel graph*[2]?

[2] A graph is series-parallel iff it does not contain a subgraph *homeomorphic* to K_4.

CHAPTER 5

Planar Support for Axis-aligned Non-piercing Rectangles

Abstract

Given n points and m *axis-parallel non-piercing rectangles* in $\mathbb{R}^2$, we present an algorithm to compute a planar support for the primal hypergraph in time $O(n \log^2 n + (n+m)\log m)$. This improves the running time over the earlier work of Raman and Ray [Rajiv Raman, Saurabh Ray: Constructing Planar Support for Non-Piercing Regions. Discret. Comput. Geom. 64(3): 1098-1122 (2020)] who gave an $O(m^2(\min\{m^3, mn\} + n))$ time algorithm to construct a planar support for n points and m non-piercing regions in the plane. Our algorithm, though for a restrictive setting, is 3-fold in its robustness: the algorithm is simple and significantly faster than the earlier work, and it gives a straight-line embedding of the support where each edge lies entirely in all the rectangles containing its end points. We also show that for a family of axis-parallel rectangles, if each point in the plane is contained in at most k pairwise *crossing* rectangles, then we can obtain a support as the union of k planar graphs. The results discussed in this chapter originate from the following joint work.
"A Fast Algorithm for Computing a Planar Support for Non-Piercing Rectangles".

TABLE OF CONTENTS

§ 5.1 Introduction

In this chapter, we construct support for geometric hypergraphs defined in the plane. We consider only primal hypergraphs and thus, by a support we mean a primal support. We define the problem statement later in this section.

So far, there are very few tools or techniques to construct a support with desired properties (e.g., planarity), even in the geometric setting. Recall that a family of simply connected regions $\mathcal{R}$, each of whose boundary is defined by a simple Jordan curve, is called *non-piercing* if for every pair of regions $R, R' \in \mathcal{R}$, both $R \setminus R'$ and $R' \setminus R$ are connected. This chapter presents a fast algorithm to construct a planar support for a restricted setting, namely, hypergraphs defined by axis-parallel rectangles that are non-piercing. This may seem rather restrictive. However, even if we allow each rectangle to belong to at most one piercing pair of rectangles, it is not difficult to construct examples where for any $r \geq 3$, any support must have $K_{r,r}$ as a topological minor. To see this, consider a geometric drawing of $K_{r,r}$ in the usual manner, i.e., the two partite sets on two vertical lines, and the edges as straight-line segments. Replace each edge of the graph by a long path, and then replace each edge along each path by a small rectangle that contains exactly two points. Where the edges cross, a pair of rectangles corresponding to each edge crosses there. Since each rectangle contains two points, it leaves us no choice as to the edges we can add. It is easy to see that the resulting support contains $K_{r,r}$ as a topological minor. Further, even for this restricted problem, the analysis of our algorithm is highly non-trivial, and we hope that the tools introduced in this chapter will be of wider interest.

Raman and Ray [RR20], showed that the hypergraph defined by non-piercing regions in the plane admits a planar support. Their proof implies an $O(m^2(\min\{m^3, mn\}+n))$ time algorithm to compute a planar support where m is the number of regions and n is the number of points in the arrangement of the regions. While their algorithm produces a plane embedding, the edges may, in general, be arbitrarily complicated curves, i.e., they may have an arbitrary number of bends. It can be shown that if the non-piercing regions are convex, then there exists an embedding of the planar support with straight edges, but it is not clear how to find such an embedding efficiently.

We present a simple and fast algorithm for drawing plane supports with straight-line edges for non-piercing rectangles. More precisely, the following is the problem definition:

Support for axis-parallel non-piercing rectangles:
Input: A set of m axis-parallel non-piercing rectangles $\mathcal{R}$ and a set P of n points in $\mathbb{R}^2$.
Output: A plane graph G on P such that for each $R \in \mathcal{R}$, $G[R \cap P]$, namely the induced subgraph on the points in $R \cap P$, is connected.

> **Note 5.1.1.** As mentioned at the start of this section, we only deal with primal support in this chapter. It should be noted that, unlike for the primal setting in previous chapters, we don't need a 2-coloring of the vertices for the problem defined above since the colored version of the problem is equivalent to the problem we defined.

Our algorithm runs in $O(n \log^2 n + (n+m) \log m)$ time, and can be easily implemented using existing data structures. The embedding computed by our algorithm not only has straight-line edges but also for each edge e, the axis-parallel rectangle with e as the diagonal does not contain any other point of P – this makes the visualization cleaner.

In order to develop a faster algorithm, we need to find a new construction (different from [RR20]), and the proof of correctness for this construction is not so straightforward. We use a sweep line algorithm. However, at any point in time, it is not possible to have the invariant that the current graph is a support for the portions of the rectangles that lie to the left of the sweep line. Instead, we show that certain *slabs* (see Section 5.3.1 for the formal definition) within each rectangle induce distinct components of the graph, and only after we sweep over a rectangle completely do we finally have the property that the set of points in that rectangle induces a connected subgraph.

▷ Organization of the chapter.

The rest of the chapter is organized as follows. We start with Section 5.2 where we present preliminary notions required for our algorithm. In Section 5.3, we present a fast algorithm to construct a planar support. We show in Section 5.3.1 the correctness of our algorithm, i.e., it does compute a planar support. We present the implementation details in Section 5.3.2. We extend the result to a general family of axis-parallel rectangles in Section 5.4, and conclude in Section 5.5 with some open questions.

§ 5.2 Preliminaries

Let $\mathcal{R} = \{R_1, \ldots, R_m\}$ denote a set of axis-parallel rectangles and let $P = \{p_1, \ldots, p_n\}$ denote a set of points in the plane. We assume that the rectangles and points are in *general position*, i.e., the points in P have distinct x and y coordinates, and the boundaries of any two rectangles in $\mathcal{R}$ are defined by distinct x-coordinates and distinct y-coordinates. Further, we assume that no point in P lies on the boundary of a rectangle in $\mathcal{R}$. For a point $p \in P$, we use $x(p)$ and $y(p)$ to denote respectively the x-coordinate and y-coordinates of p. In the following, we define some notions that we will be using throughout this chapter.

> **Definition 5.1** (Piercing, discrete piercing)**.** A rectangle R' is said to *pierce* a rectangle R if $R \setminus R'$ consists of two connected components. A rectangle R' *discretely pierces* a rectangle R if R' pierces R and each component of $R \setminus R'$ contains a point of P.

Note that while piercing is a symmetric relation, discrete piercing is not.

> **Definition 5.2** (Non-piercing)**.** A collection $\mathcal{R}$ of rectangles is *non-piercing* if no pair of rectangles $R, R' \in \mathcal{R}$ pierce each other.

'L'-shaped edge. We construct a drawing of a support graph G on P using 'L'-shaped edges of type: $\urcorner$ or $\lrcorner$. Henceforth, the term *edge* will mean one of the two 'L'-shaped edges joining two points. The embedded graph may not be planar due to the overlap of the edges along their horizontal/vertices segments. However, as we show, G satisfies the additional property that for each edge, the axis-parallel rectangle defined by the edge has no points of P in its interior (formal definition below), and that no pair of edges cross. Consequently, replacing each edge with the straight-line segment joining its end-points, yields a plane embedding of G.

> **Definition 5.3** (Delaunay edge)**.** For an edge between points $p, q \in P$, let $R(pq)$ denote the axis-parallel rectangle with diagonally opposite corners p and q. The edge pq is a *Delaunay edge* if the interior of $R(pq)$ does not contain a point of P.

> **Definition 5.4** (Valid edge)**.** We say that an edge pq (discretely) pierces a rectangle R if $R \setminus \{pq\}$ consists of two regions, and each region contains a point of P. An edge pq is said to be *valid* if it does not discretely pierce any rectangle $R \in \mathcal{R}$, and does not cross any existing edge.

For an edge pq, we use $h(pq)$ for the horizontal segment of pq, and $v(pq)$ for the vertical segment of pq.

A curve γ in the plane is said to be x-monotone if any vertical line parallel to the y-axis does not intersect γ more than once. We modify this definition slightly for our purpose, and we define the following:

> **Definition 5.5** (Monotone path)**.** A path π in the plane formed by a sequence of L-shaped edges $\urcorner$, or $\lrcorner$ is said to be x-monotone, if any vertical line intersects π in at most one vertical segment (which may in some cases be a single point).

Point above(below) Path. Let π be a path and q be a point not on the path. We say that "q lies above π" if ℓ_q, the vertical line through q intersects π at point(s) below q. We define the notion that "q lies below π" analogously. Note that these notions are defined only if ℓ_q intersects π.

Left(Right)-Neighbor, Left(Right)-Adjacent. For a point $q \in P$ and a set $P' \subseteq P$, the *right-neighbor* of q in P' is q_1, where $q_1 = \text{argmin}_{q' \in P'}\{x(q') : x(q') > x(q)\}$. The *left-neighbor* of q in P' is defined similarly, i.e., q_0 is the left-neighbor of q, where $q_0 = \text{argmax}_{q' \in P'}\{x(q') : x(q') < x(q)\}$. Note that being a left- or right-neighbor is a *geometric notion*, and not related to the support graph we construct. We use the term *left-adjacent* to refer to the neighbors of q in a plane graph G that lie to the left of q. The term *right-adjacent* is defined analogously.

§ 5.3 Algorithm

In this section, we present an algorithm to compute a planar support for the hypergraph defined by points and non-piercing axis-parallel rectangles in $\mathbb{R}^2$: Perform a left-to-right vertical line sweep and at each input point encountered, add all possible *valid Delaunay edges* to previous points. The algorithm, presented as Algorithm 1, draws edges having shapes in $\{\urcorner, \lrcorner\}$. We prove correctness of Algorithm 1 in Section 5.3.1, and show how it can be implemented to run in $O(n \log^2 n + (n+m) \log m))$ time in Section 5.3.2.

Input: A set P of points, a set $\mathcal{R}$ of non-piercing axis-parallel rectangles in $\mathbb{R}^2$.
Output: Embedded Planar Support $G = (P, E)$
Order P in increasing order of *x-coordinates*: $(p_1, \ldots, p_n)$
$E = \emptyset$
for each point p_i in sorted order, $i \in \{2, 3, \ldots, n\}$ **do**
| $E = E \cup \{e_{ij} = p_i p_j \mid j < i,$ and e_{ij} is a *valid Delaunay edge*.$\}$
end

Algorithm 1: Planar Support.

Algorithm 1 outputs a graph G on P embedded in $\mathbb{R}^2$, whose edges are valid Delaunay edges of type $\{\urcorner, \lrcorner\}$. Replacing each Delaunay edge $\{p, q\}$ by the diagonal of $R(pq)$ yields a plane embedding of G. Fig. 5.1 shows an illustration of the algorithm with point set $p, q, \ldots, v$, and rectangles R_1, R_2, R_3. The edges with both end points contained in rectangle R_1, are shown. This includes edges – $qp, rp, rq, tr, tq, ur, vr, vt$.

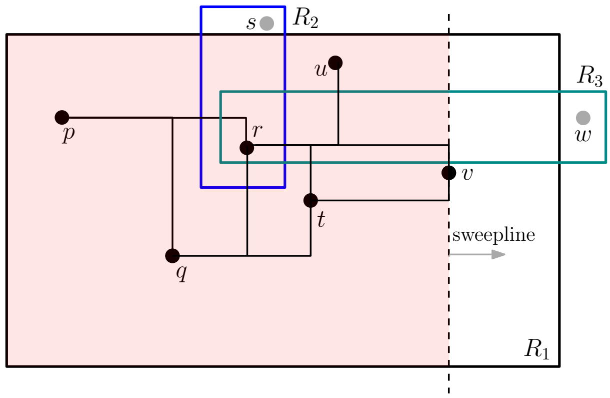


Figure 5.1: Illustration of Algorithm 1. Edges with both end points in rectangle R_1 are shown.

Note that the algorithm does not add edge tp since it is not Delaunay as the axis-parallel rectangle with diagonal pt contains the point r. Moreover, the edge tp would also discretely pierce the rectangle R_2.

§ 5.3.1 Correctness of algorithm

In this section, we show that the graph G on vertex set P constructed by Algorithm 1 is a support graph for the rectangles in $\mathcal{R}$, and this is sufficient as planarity follows directly by construction. The proof is technical, and we start with some necessary notations required for the proof.

▷ Coordinates of a rectangle.

For a rectangle R, we denote the y-coordinates of its lower and upper horizontal sides by $y_-(R)$ and $y_+(R)$, respectively. Similarly, $x_-(R)$ and $x_+(R)$ denote respectively the x-coordinates of the left and right vertical sides. We denote the vertical line through any point p by ℓ_p.

We use $\textsc{Piece}(R, H)$ to denote the rectangle $R \cap H$ for a halfplane H defined by a vertical line. We abuse notation and use $\textsc{Piece}(R, p)$ to denote the rectangle $R \cap H_-(\ell_p)$, the intersection of R with the left half-space defined by the vertical line through the point p.

We also use the notation $R[x_-, x_+]$ to denote the sub-rectangle of rectangle R, that lies between x-coordinates x_- and x_+. Similarly, we use $R[y_-, y_+]$ to denote the sub-rectangle of R that lies between the y-coordinates y_- and y_+.

▷ Boundary conditions.

To avoid boundary conditions in the definitions that follow, we add two rectangles: R_{top} above all rectangles in $\mathcal{R}$, and R_{bot} below all rectangles in $\mathcal{R}$, that is $y_-(R_{top}) >$

$\max_{R\in\mathcal{R}} y_+(R)$, and $y_+(R_{bot}) < \min_{R\in\mathcal{R}} y_-(R)$. The rectangles R_{top}, and R_{bot} span the width of all rectangles, i.e., $x_-(R_{top}) = x_-(R_{bot}) < \min_{R\in\mathcal{R}} x_-(R)$, and $x_+(R_{top}) = x_+(R_{bot}) > \max_{R\in\mathcal{R}} x_+(R)$. We add two points $P_{top} = \{p_1^+, p_2^+\}$ to the interior of R_{top}, and two points $P_{bot} = \{p_1^-, p_2^-\}$ to the interior of R_{bot}, such that $x(p_1^+) = x(p_1^-) < \min_{p\in P} x(p)$, and $x(p_2^+) = x(p_2^-) > \max_{p\in P} x(p)$. Let $\mathcal{R}' = \mathcal{R} \cup \{R_{top}, R_{bot}\}$, and $P' = P \cup P_{top} \cup P_{bot}$. For ease of notation, we simply use $\mathcal{R}$ and P to denote $\mathcal{R}'$ and P' respectively, and implicitly assume the existence of $R_{top}, R_{bot}, P_{top}$ and P_{bot}.

▷ ACTIVE rectangles.

For a vertical segment s, a rectangle $R \in \mathcal{R}'$ is said to be *active* at s if it is either discretely pierced by s, i.e., $R\backslash s$ is not connected and each of the two components contains a point of P, or there is a point of $P \cap s$ in R. We denote the set of all active rectangles at s by $\textsc{Active}(s)$. For a point $p \in P \cap s$, we define $\textsc{Contain}(s, p)$ to be the set of rectangles in $\textsc{Active}(s)$ that contains the point p. We define $\textsc{Above}(s, p)$ to be the set of rectangles in $\textsc{Active}(s)$ that lie strictly above p, i.e., $\textsc{Above}(s, p) = \{R \in \textsc{Active}(s) : y_-(R) > y(p)\}$. Similarly, $\textsc{Below}(s, p) = \{R \in \textsc{Active}(s) : y_+(R) < y(p)\}$. It follows that for any point $p \in s$, $\textsc{Active}(s) = \textsc{Contain}(s, p) \sqcup \textsc{Above}(s, p) \sqcup \textsc{Below}(s, p)$, where $\sqcup$ denotes the disjoint union.

Note that for the vertical line ℓ_p through $p \in P$, $\textsc{Active}(\ell_p) \neq \emptyset$, as $\textsc{Active}(\ell_p)$ contains the rectangles R_{top} and R_{bot}. Similarly, $\textsc{Above}(\ell_p, p) \neq \emptyset$ and $\textsc{Below}(\ell_p, p) \neq \emptyset$. Abusing notations slightly, we write $\textsc{Active}(p)$ instead of $\textsc{Active}(\ell_p)$, and likewise with $\textsc{Contain}(\cdot)$, $\textsc{Above}(\cdot)$ and $\textsc{Below}(\cdot)$.

▷ Barriers at a point.

For a point $p \in P$, we now introduce the notion of *barriers*. Any active rectangle R' in $\textsc{Above}(p)$ prevents a valid Delaunay edge incident on p from being incident to a point to the left of p above $y_+(R')$, as such an edge would discretely pierce R'. Hence, among all rectangles $R' \in \textsc{Above}(p)$, the one with lowest $y_+(R')$ is called the *upper barrier* at p, denoted $\textsc{Ub}(p)$. Thus, $\textsc{Ub}(p) = \operatorname{argmin}_{R'\in\textsc{Above}(p)} y_+(R')$. Similarly, we define the *lower barrier* at p, $\textsc{Lb}(p) = \operatorname{argmax}_{R'\in\textsc{Below}(p)} y_-(R')$. Note that $\textsc{Ub}(p)$ and $\textsc{Lb}(p)$ exist for any $p \in P$ since $\textsc{Above}(p)$ and $\textsc{Below}(p)$ are non-empty.

While the rectangles in $\mathcal{R}'$ are non-piercing, a rectangle $R' \in \textsc{Active}(p)$ can be discretely pierced by $\textsc{Piece}(R, p)$. We thus define the *upper piercing barrier* $\textsc{Upb}(R, p)$ as the rectangle $R' \in \textsc{Above}(p)$ with the lowest $y_+(R')$ that is pierced by $\textsc{Piece}(R, p)$, and we define the *lower piercing barrier* $\textsc{Lpb}(R, p)$ analogously. That is,

$$\textsc{Upb}(R,p) = \operatorname*{argmin}_{\substack{R'\in\textsc{Above}(p)\\ \textsc{Piece}(R,p)\text{ pierces }R'}} y_+(R') \quad \text{and} \quad \textsc{Lpb}(R,p) = \operatorname*{argmax}_{\substack{R'\in\textsc{Below}(p)\\ \textsc{Piece}(R,p)\text{ pierces }R'}} y_-(R')$$

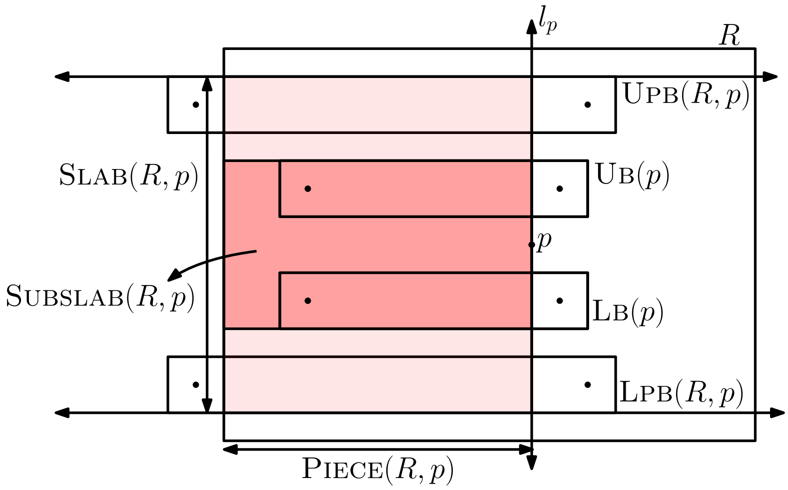


Figure 5.2: The figure above shows $\textsc{Ub}(p)$, $\textsc{Lb}(p)$, and the upper and lower piercing barriers $\textsc{Lpb}(R,p)$ and $\textsc{Upb}(R,p)$ of $\textsc{Piece}(R,p)$. The slab $\textsc{Slab}(R,p)$ containing p defined by $\textsc{Upb}(R,p)$ and $\textsc{Lpb}(R,p)$ is shaded. The dark grey part shows the $\textsc{Subslab}(R,p)$.

▷ Partitioning a rectangle into SLAB(s).

For a point $p \in P$ and a rectangle $R \in \textsc{Contain}(p)$, if $\textsc{Upb}(R,p)$ or $\textsc{Lpb}(R,p)$ exist, then the horizontal line containing $y_+(\textsc{Upb}(R,p))$ together with the horizontal line containing $y_-(\textsc{Lpb}(R,p))$ naturally split $\textsc{Piece}(R,p)$ into at most three sub-rectangles called *slabs*. The point p lies in exactly one of these slabs, denoted $\textsc{Slab}(R,p)$. Thus, $\textsc{Slab}(R,p)$ is the sub-rectangle of R whose left and right-vertical sides are respectively defined by $x_-(R)$ and ℓ_p, and the upper and lower sides are respectively defined by

$$y_+(\textsc{Slab}(R,p)) = \begin{cases} y_+(\textsc{Upb}(R,p)), & \text{if } \textsc{Upb}(R,p) \text{ exists} \\ y_+(R), & \text{otherwise} \end{cases}$$

and similarly,

$$y_-(\textsc{Slab}(R,p)) = \begin{cases} y_-(\textsc{Lpb}(R,p)), & \text{if } \textsc{Lpb}(R,p) \text{ exists} \\ y_-(R), & \text{otherwise} \end{cases}$$

By definition, for a point p and $R \in \textsc{Contain}(p)$, if $\textsc{Upb}(R,p)$ exists, then $y_+(\textsc{Upb}(R,p)) \geq y_+(\textsc{Ub}(p))$. Similarly, if $\textsc{Lpb}(R,p)$ exists, then $y_-(\textsc{Lpb}(R,p)) \leq y_-(\textsc{Lb}(p))$. Thus, $y_+(\textsc{Ub}(p))$ and $y_-(\textsc{Lb}(p))$ together split $\textsc{Slab}(R,p)$ further into at most 3 sub-rectangles called *sub-slabs* whose vertical sides coincide with the vertical sides of $\textsc{Slab}(R,p)$, and the horizontal sides are defined by $y_+(\textsc{Ub}(p))$ and $y_-(\textsc{Lb}(p))$. Let $\textsc{Subslab}(R,p)$ denote the sub-slab containing p. Fig. 5.2 illustrates the notions defined thus far. Note that the left-adjacent vertices of p in G that are contained in R, only lie in $\textsc{Subslab}(R,p)$.

Now we are ready to prove the correctness of Algorithm 1. We start with a proof strategy below.

▷ Proof strategy.

To prove that the graph G constructed by Algorithm 1 is a support for $\mathcal{R}$, we proceed in two steps. First (and the part that requires most of the work) we show that for each $R \in \mathcal{R}$ and $p \in P \cap R$, the subgraph of G induced by the points in $\textsc{Slab}(R, p)$ is connected. Second, we show that if p is the rightmost point in R, then $\textsc{Slab}(R, p)$ contains *all* points in $R \cap P$ which, by the first part, is connected.

When processing a point p, Algorithm 1 only adds valid Delaunay edges from p to points to its left. That is, we only add edges to a subset of points in $\textsc{SubSlab}(R, p)$. To show that $\textsc{Slab}(R, p)$ is connected, one approach could be to show that the $\textsc{Slab}(R, p)$ is covered by sub-slabs defined by points in $\textsc{Slab}(R, p)$, adjacent sub-slabs share a point of P, and that points in a sub-slab induce a connected subgraph. Unfortunately, this is not true, and we require a finer partition of a slab. We proceed as follows: First, we define a sequence of sub-rectangles of $\textsc{Slab}(R, p)$ called *strips*, denoted $\textsc{Strip}(R, p, i)$ for $i \in \{-t, \ldots, k\}$, where the strips that lie above p have positive indices, the strips that lie below p have negative indices, and the unique strip that contains p has index 0. Further, each strip shares its vertical sides with $\textsc{Slab}(R, p)$. In the following, since R and p are fixed, we refer to $\textsc{Strip}(R, p, i)$ as $\textsc{Strip}_i$. We define the strips so that they satisfy the following conditions:

- (C_1) Each strip is contained in the slab, i.e, $\textsc{Strip}_i \subseteq \textsc{Slab}(R, p)$ for each $i \in \{-t, \ldots, k\}$.
- (C_2) The union of strips cover the slab, i.e., $\textsc{Slab}(R, p) \subseteq \cup_{i=-t}^{k} \textsc{Strip}_i$, and
- (C_3) Any two consecutive strips contain a point of P in their intersection, i.e., $\textsc{Strip}_i \cap \textsc{Strip}_{i-1} \cap P \neq \emptyset$ for all $i \in \{-t+1, \ldots, k\}$. Consequently, each strip contains a point of P.

In order to prove that $\textsc{Slab}(R, p)$ is connected, we describe below a strategy that does not quite work but, as we show later, can be fixed.

Let $\textsc{Strip}_i \cap P = P_i$. By condition (C_3), $P_i \neq \emptyset$ for any $i \in \{-t, \ldots, k\}$. For a strip $\textsc{Strip}_i$, let p_i denote the rightmost point in it. Let us assume[1] for now that for each $i \in \{-t, \ldots, k\}$, and each point $q \in P_i$, there is a path from q to p_i that lies entirely in $\textsc{Strip}_i$. Now, consider an arbitrary point $q \in \textsc{Slab}(R, p)$. By condition (C_2) each point in $P \cap \textsc{Slab}(R, p)$ is contained in at least one strip. Therefore, $q \in \textsc{Strip}_i$ for some $i \in \{-t, \ldots, k\}$. By our assumption, there is a path π_i^1 from q to p_i that lies entirely in $\textsc{Strip}_i$. If $i \geq 0$ (a symmetric argument works when $i < 0$), since condition (C_3) implies consecutive strips intersect at a point in P, there is a path π_i^2 from p_i to a point $q' \in P_i \cap P_{i-1}$ that lies entirely in $\textsc{Strip}_i$. Again, by our assumption, there is a path π_{i-1}^1 from q' to p_{i-1} that lies entirely in $\textsc{Strip}_{i-1}$. Repeating the argument above with

[1] This assumption is incorrect but will be remedied later.

$i-1, i-2, \ldots$, until $i=0$, and concatenating the paths $\pi_i^1, \pi_i^2, \pi_{i-1}^1, \ldots$, we obtain a path π from q to p, each sub-path of which is a path from a point in a strip to the rightmost point in that strip such that each point in the path lies entirely in the strip. By condition (C_1), $\text{STRIP}_i \subseteq \text{SLAB}(R,p)$ for each $i \in \{-t, \ldots, k\}$. Therefore, π lies entirely in $\text{SLAB}(R,p)$. Since q was arbitrary, this implies that $\text{SLAB}(R,p)$ is connected.

▷ Construction of STRIPS.

Consider a slab $\text{SLAB}(R,p)$ corresponding to a rectangle $R \in \mathcal{R}$ and a point $p \in P \cap R$. The strips corresponding to $\text{SLAB}(R,p)$ are defined as follows: Let s denote the open segment of ℓ_p between p and $y_+(\text{SLAB}(R,p))$ where ℓ_p is the vertical line through p. Let $\mathcal{R}_s = (R_0, \ldots, R_h)$ be the rectangles in $\text{ACTIVE}(s)$ ordered by their upper sides, i.e., $y_+(R_i) < y_+(R_j)$, for $0 \le i < j \le h$. Similarly, let s' denote the open segment of ℓ_p between p and $y_-(\text{SLAB}(R,p))$ and let $\mathcal{R}_{s'} = (R'_0, \ldots, R'_{h'})$ denote the rectangles in $\text{ACTIVE}(s')$ ordered by their lower sides $y_-(R'_i) > y_-(R'_j)$ for $0 \le i < j \le h'$.

We define $\text{STRIP}_0 = \text{SLAB}(R,p)[y_-(R'_0), y_+(R_0)]$, if $\text{ACTIVE}(s) \neq \emptyset$ and $\text{ACTIVE}(s') \neq \emptyset$. If $\text{ACTIVE}(s) = \emptyset$, we set $y_+(\text{STRIP}_0) = y_+(\text{SLAB}(R,p))$. Similarly, if $\text{ACTIVE}(s') = \emptyset$, we set $y_-(\text{STRIP}_0) = y_-(\text{SLAB}(R,p))$. We set $p_0 = p$. Having defined STRIP_0, we set $\mathcal{R}_s = \mathcal{R}_s \setminus R_0$ and $\mathcal{R}_{s'} = \mathcal{R}_{s'} \setminus R'_0$.

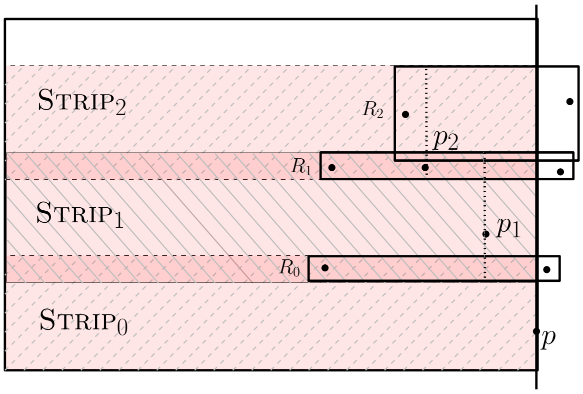


Figure 5.3: The figure shows the construction of the strips STRIP_0, STRIP_1 and STRIP_2. The vertical line segment through p_i, $i \in \{1,2\}$ shows that p_i is the rightmost point among the points in STRIP_i.

For $i > 0$, having constructed STRIP_j for $j = 0, \ldots, i-1$, we do the following while $\mathcal{R}_s \neq \emptyset$: Let $S_i = \text{argmin}\{y_-(R') : R' \in \mathcal{R}_s\}$, and let $y_- = y_-(S_i)$. Let $R_i = \text{argmin}\{y_+(R') : R' \in \mathcal{R}_s : y_-(R') > y_-\}$, and let $y_+ = \min\{y_+(\text{SLAB}(R,p)), y_+(R_i)\}$. Set $y_-(\text{STRIP}_i) = y_-$ and $y_+(\text{STRIP}_i) = y_+$. Let $p_i = \text{argmax}\{x(p') : p' \in P \cap \text{SLAB}(R,p) : y_- < y(p') < y_+\}$. Note that p_i exists since $S_i \in \text{ACTIVE}(s)$. Set $\mathcal{R}_s = \mathcal{R}_s \setminus \{R' : y_-(R') < y_-(R_i)\}$.

For $i < 0$, the construction is symmetric. Having constructed $\text{STRIP}(R,p,j)$ for j

from 0 down to $-i+1$, we do the following until $\mathcal{R}_{s'} = \emptyset$. Let $S'_i = \operatorname{argmax}\{y_+(R') : R' \in \mathcal{R}_{s'}\}$, and let $y_+ = y_+(S'_i)$. Let $R'_i = \operatorname{argmax}\{y_-(R') : R' \in \mathcal{R}_{s'}, y_+(R') < y_+\}$. Let $y_- = \max\{y_-(R'_i), y_-(\textsc{Slab}(R,p))\}$. Set $y_-(\textsc{Strip}_i) = y_-$ and $y_+(\textsc{Strip}_i) = y_+$. Let $p_i = \operatorname{argmax}\{x(p') : p' \in P \cap \textsc{Slab}(R,p), y_- < y(p') < y_+\}$. Again, p_i exists since $S'_i \in \textsc{Active}(s')$. Set $\mathcal{R}_{s'} = \mathcal{R}_{s'} \setminus \{R' \in \mathcal{R}_{s'} : y_+(R') > y_+(R'_i)\}$. Fig. 5.3 illustrates the construction of the strips.

Proposition 5.1. For each $i \in \{-t,\dots,k\}$, the point p_i and the corresponding $\textsc{Strip}_i$ satisfy that $y_-(\textsc{Lb}(p_i)) \le y_-(\textsc{Strip}_i) < y_+(\textsc{Strip}_i) \le y_+(\textsc{Ub}(p_i))$.

Proof. Fix $i \in \{-t,\dots,k\}$ and assume $i \ge 0$. For $i < 0$, the proof is symmetric. Since $p_i \in \textsc{Strip}_i$ and by the definition of the lower barrier, $y_+(\textsc{Lb}(p_i)) < y(p_i) < y_+(\textsc{Strip}_i)$. If $y_-(\textsc{Lb}(p_i)) > y_-(\textsc{Strip}_i)$, since $\textsc{Lb}(p_i) \in \mathcal{R}_s$ and $y_+(\textsc{Strip}_i) \le \min\{y_+(R') \in \mathcal{R}_s : R' \in \mathcal{R}_s \text{ and } y_-(R') > y_-(\textsc{Strip}_i)\}$, it implies $y_+(\textsc{Strip}_i) \le y_+(\textsc{Lb}(p_i))$, contradicting $p_i \in \textsc{Strip}_i$.

Now we argue about $\textsc{Ub}(p_i)$. If $y_+(\textsc{Ub}(p_i)) > y_+(\textsc{Slab}(R,p))$, since $y_+(\textsc{Slab}(R,p)) \ge y_+(\textsc{Strip}_i)$, we have $y_+(\textsc{Ub}(p_i)) \ge y_+(\textsc{Strip}_i)$. Otherwise, we have $\textsc{Ub}(p_i) \in \mathcal{R}_s$. Since $p_i \in \textsc{Strip}_i$ and by definition of the upper barrier, we have $y_-(\textsc{Ub}(p_i)) > y(p_i) > y_-(\textsc{Strip}_i)$. Since $y_+(\textsc{Strip}_i) \le \min\{y_+(R') : R' \in \mathcal{R}_s \text{ and } y_-(R') > y_-(\textsc{Strip}_i)\}$, it follows that $y_+(\textsc{Strip}_i) \le y_+(\textsc{Ub}(p_i))$. ■

Our next lemma shows that the strips we constructed satisfy the required conditions (C_1) to (C_3).

Lemma 5.1. The strips constructed as above satisfy the following conditions: (i) $\textsc{Strip}_i \subseteq \textsc{Slab}(R,p)$ for each $i \in \{-t,\dots,k\}$ (ii) $\textsc{Slab}(R,p) = \cup_{i=-t}^{k}\textsc{Strip}_i$, and (iii) $\textsc{Strip}_i \cap \textsc{Strip}_{i-1} \cap P \neq \emptyset$ for all $i \in \{-t+1,\dots,k\}$.

Proof. Items (i) and (ii) follow directly by construction. For (iii), note that adjacent strips contain a piece of an active rectangle and hence their intersection contains a point of P. ■

Unfortunately, our assumption that every point in a strip has a path to its rightmost point in the strip is not correct. To see this, consider a strip that is pierced by a rectangle R', whose intersection with the strip does not contain a point of P, but has points above and below the strip (see Fig. 5.4). Therefore, a point in the strip that lies to the left of R' cannot have a path to p_i that lies in the strip unless some of its edge is allowed to discretely pierce R'.

In order to remedy this situation, we introduce the notion of a *corridor*. A corridor corresponding to a strip is a region of $\textsc{Slab}(R,p)$ that contains all points in the

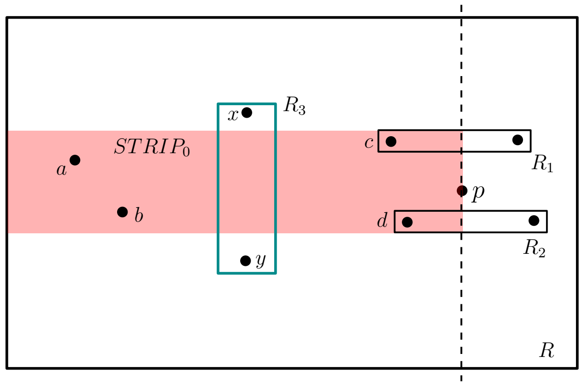


Figure 5.4: Rectangle R_3 hinders a path between a and p to lie inside the strip (shaded red), and thus points in a strip need not form a connected graph.

strip, and such that each point in the strip has a path to the rightmost point in it that lies entirely in the corridor. Since each corridor lies in $\textsc{Slab}(R, p)$, the proof strategy discussed for strips can be suitably modified to show that $\textsc{Slab}(R, p)$ is connected.

We now define the corridors associated with each strip. Recall that $G = (P, E)$ is the plane graph constructed by Algorithm 1. For a point $q \in P$, recall that the neighbors of q in G that lie to its left are called its *left-adjacent* points. If q lies on a path π, and q' is the left-adjacent point of q on π, then we say that q' is left-adjacent to q on π. We start with the following proposition that will be useful in constructing the corridors.

Proposition 5.2. For a point $q \in P$, if $(q_1, \ldots, q_r, q_{r+1}, \ldots, q_s)$ is the sequence of its left-adjacent points in G such that $y(q_1) > \ldots > y(q_r) > y(q)$ and $y(q_{r+1}) < \ldots < y(q_s) < y(q)$. Then $x(q_i) > x(q_j)$ for $1 \leq i < j \leq r$, and $x(q_i) < x(q_j)$ for $r + 1 \leq i < j \leq s$.

Proof. This follows directly from the fact that each edge in G is Delaunay. ■

▷ Construction of $\textsc{Corridors}$.

For a strip $\textsc{Strip}_i$, we define its corresponding corridor, namely $\textsc{Corridor}_i$ as follows: A corridor is the region of $\textsc{Slab}(R, p)$ bounded by two paths: an *upper path* π_i^u, and a *lower path* π_i^ℓ, which we define below.

The upper path $\pi_i^u = (q_0, q_1, q_2, \ldots,)$ is constructed by starting with $j = 0, q_0 = p_i$, and repeating (1) set $q_{j+1} \leftarrow q'$ where q' has the highest $y(q')$ among the left-adjacent points of q_j. (2) $j \leftarrow j + 1$. We stop when we cannot find such a q' for the current q_j in G that lies in $\textsc{Slab}(R, p)$, where we complete the path by following the horizontal segment of the edge to the left-adjacent vertex q_{j+1} of q_j with the highest y-coordinate. Thus, the last vertex of π_i^u possibly does not lie in $\textsc{Slab}(R, p)$.

The lower path $\pi_i^\ell = (q'_0, \ldots,)$ is constructed similarly. For $j = 0$, set $q'_0 = p_i$, and repeating (1) set $q'_{j+1} \leftarrow q'$, where q' has the *lowest* $y(q')$ among the left-adjacent points of q'_j. We stop when we cannot find such a q' in $\textsc{Slab}(R, p)$ for the current q'_j, where we complete the path by following the horizontal segment of the edge from q'_j its left-adjacent vertex q'_{j+1} with the smallest y-coordinate. Thus, the last vertex of π_i^ℓ possibly does not lie in $\textsc{Slab}(R, p)$.

$\textsc{Corridor}_i$ is the region of $\textsc{Slab}(R, p)$ that lies between the upper and lower paths π_i^u and π_i^ℓ. Fig. 5.5 shows a corridor corresponding to a strip. We start with some basic observations about the corridors thus constructed.

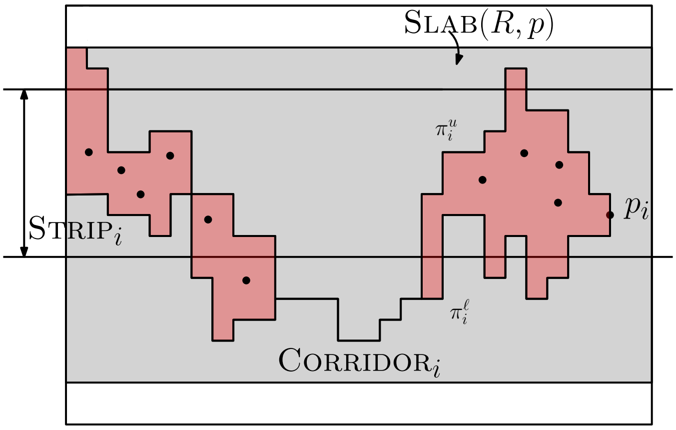


Figure 5.5: The figure above shows the strip $\textsc{Strip}_i$, the slab $\textsc{Slab}(R, p)$ in grey, and the corridor $\textsc{Corridor}_i$ as the region shaded in red between π_i^u and π_i^ℓ.

Proposition 5.3. For $i \in \{-t, \ldots, k\}$, the paths π_i^u and π_i^ℓ are x-monotone.

Proof. This follows directly by construction since at each step we augment the path by adding to it, the left-adjacent neighbor to the current vertex of the path. ∎

The edges in graph G constructed by Algorithm 1 do not cross. We say that two paths π_1 and π_2 in G *cross* if there is an x-coordinate at which π_1 lies above π_2, and an x-coordinate at which π_2 lies above π_1. We show the following.

Proposition 5.4. For each $i \in \{-t, \ldots, k\}$, the path π_i^l does not lie above the path π_i^u. Also, π_i^ℓ and π_i^u do not cross.

Proof. Let $\pi_i^u = (q_0, q_1, \ldots, q_r)$ and $\pi_i^\ell = (q'_0, \ldots, q'_s)$, where $q_0 = q'_0 = p_i$. Since q_1 is the left-adjacent of p_i in $\textsc{Slab}(R, p)$ with highest y-coordinate and q'_1 is the left-adjacent point of p_i with lowest y-coordinate, $y(q'_1) \leq y(q_1)$. Thus, π_i^ℓ does not lie above π_i^u at $x(p_i)$. If at some x-coordinate x', π_i^ℓ lies above π_i^u, then the paths must have crossed to the right of x'.

Let $q_i = q'_j = q$ be a point of P common to π_i^u and π_i^ℓ lying to the left of p_i. Again, since $q_{i+1} = \arg\max\{y(q'') : q'' \in \text{SLAB}(R, p) \cap P, x(q'') < x(q), \{q, q''\} \in E(G)\}$, and $q'_{j+1} = \arg\min\{y(q'') : q'' \in \text{SLAB}(R, p) \cap P, x(q'') < x(q), \{q, q''\} \in E(G)\}$, it follows that $y(q'_{j+1}) \leq y(q_{i+1})$. Hence, the paths do not cross, and since π_i^ℓ does not lie above π_i^u at $x(p_i)$, it does not do so at any x-coordinate to the left of p_i either. ■

Recall that left-neighbor of a point q in a set P' is the point $p' = \text{argmax}_{p'' \in P'}\, x(p'') < x(q)$. The right-neighbor is defined similarly. Note that the left and right neighbors are defined geometrically, and they may not be adjacent to q in the graph G. For a point $q \in P$ and $i \in \{-t, \ldots, k\}$, we let r_0 and r_1 denote respectively, the left- and right-neighbors of q on π_i^u. Similarly, we let r'_0 and r'_1 denote respectively, the left- and right-neighbors of q on π_i^ℓ. We show the following.

Proposition 5.5. For each $i \in \{-t, \ldots, k\}$ and $q \in P$, if q lies above π_i^u, then $y(q) > \max\{y(r_0), y(r_1)\}$. Similarly, if q lies below π_i^ℓ, then $y(q) < \min\{y(r'_0), y(r'_1)\}$.

Proof. We prove the case when q lies above π_i^u. The other case follows by an analogous argument. By Proposition 5.3, since π_i^u is x-monotone, it follows that r_0 and r_1 are consecutive along π_i^u, and thus, $r_0 r_1$ is a valid Delaunay edge in G. If either $y(r_0) > y(q) > y(r_1)$, or $y(r_0) < y(q) < y(r_1)$, then it contradicts the fact that $r_0 r_1$ is Delaunay. Hence, $y(q) > \max\{y(r_0), y(r_1)\}$ as q lies above π_i^u. ■

We now show that the corridors constructed satisfy the required conditions. The first condition below, follows directly from construction.

Lemma 5.2. For each $i \in \{-t, \ldots, k\}$, $\text{CORRIDOR}_i \subseteq \text{SLAB}(R, p)$.

Proof. This is followed directly by the construction of the corridor since it is defined to be the region of the $\text{SLAB}(R, p)$ that lies between the upper and lower paths, namely, π_i^u and π_i^ℓ. ■

Next, we show that for each strip, its corresponding corridor contains all its points, that is, all points in $P \cap \text{STRIP}_i$ are contained between the upper and lower paths of CORRIDOR_i. Before we do that, we need the following two technical statements.

Proposition 5.6. Let $q, q' \in P$, with $x(q) < x(q')$ such that qq' is Delaunay. If $qq' \notin E(G)$, then either (i) $h(qq')$ pierces a rectangle, or crosses an existing edge, or (ii) $v(qq')$ pierces a rectangle. In particular, $v(qq')$ does not cross an existing edge.

Proof. The points in P are processed by Algorithm 1 in increasing order of their x-coordinates, and when a point is being processed, we add edges of type $\{\urcorner, \lrcorner\}$ to points to its left. Therefore, while processing q', no edge from points of P to the right of q' has been added. Hence, $v(qq')$ does cross an existing edge. ∎

Lemma 5.3. Consider $i \in \{-t, \ldots, k\}$.

1. Let $q \in \textsc{Slab}(R, p) \cap P$ such that q lies above π_i^u, and let q_1 be the right-neighbor of q on π_i^u. If qq_1 is Delaunay but not valid, then $v(qq_1)$ pierces a rectangle. In particular, $h(qq_1)$ does not pierce a rectangle or cross an edge.
2. Let $q' \in \textsc{Slab}(R, p) \cap P$ such that q' lies below π_i^ℓ, and let q_1' is the right-neighbor of q' on π_i^ℓ. If $q'q_1'$ is Delaunay but not valid, then $v(q'q_1')$ pierces a rectangle. In particular $h(q'q_1')$ does not pierce a rectangle or cross an edge.

Proof. We prove the case when q lies above π_i^u. The other case follows by an analogous argument. Since qq_1 is not valid, either the horizontal segment of qq_1 pierces a rectangle, or crosses an existing edge; or $v(qq_1)$ pierces a rectangle since by Proposition 5.6, $v(qq_1)$ does not cross an existing edge.

Let q_0 be left-adjacent to q_1 on π_i^u. By Proposition 5.5, $y(q) > \max\{y(q_0), y(q_1)\}$. Hence qq_1 is of type $\urcorner$. Suppose $h(qq_1)$ pierces a rectangle or crosses an edge of type $\urcorner$. Then, there is a point z that lies below the $h(qq_1)$. But, z cannot lie below the $h(q_0q_1)$, as that contradicts the fact that q_0q_1 is valid. Hence, z lies above $h(q_0q_1)$. This implies that z lies either in $R(q_0q_1)$ (if $y(q_0) < y(q_1)$), or z lies in $R(qq_1)$. If $z \in R(q_0q_1)$, it contradicts the fact that q_0q_1 is Delaunay. Also, $z \notin R(qq_1)$, as qq_1 is Delaunay by assumption. Therefore, $h(qq_1)$ does not pierce a rectangle, or crosses an edge of type $\urcorner$. If $h(qq_1)$ crossed an edge e of type $\lrcorner$, then either qq_1 is not Delaunay, violating our assumption, or e is not Delaunay, a contradiction. ∎

Lemma 5.4. For each $i \in \{-t, \ldots, k\}$, $P_i \subseteq \textsc{Corridor}_i$, where $P_i = P \cap \textsc{Strip}_i$.

Proof. Suppose $P_i \setminus \textsc{Corridor}_i \neq \emptyset$. By Proposition 5.4, any such point either lies above π_i^u or below π_i^ℓ. We assume the former. The latter is followed by an analogous argument. Let $P_i' = \{q \in \textsc{Slab}(R, p) : y(q) < y_+(\textsc{Strip}_i)$ and q lies above $\pi_i^u\}$. It suffices to show that $P_i' = \emptyset$. For the sake of contradiction, suppose $P_i' \neq \emptyset$.

We impose the following partial order on P_i: for $a, a' \in P_i$, $a \prec a' \Leftrightarrow x(a) > x(a') \wedge y(a) < y(a')$. Let q be a minimal element in P_i' according to $\prec$. In the following, when we refer to a *minimal* element, we implicitly assume this partial order.

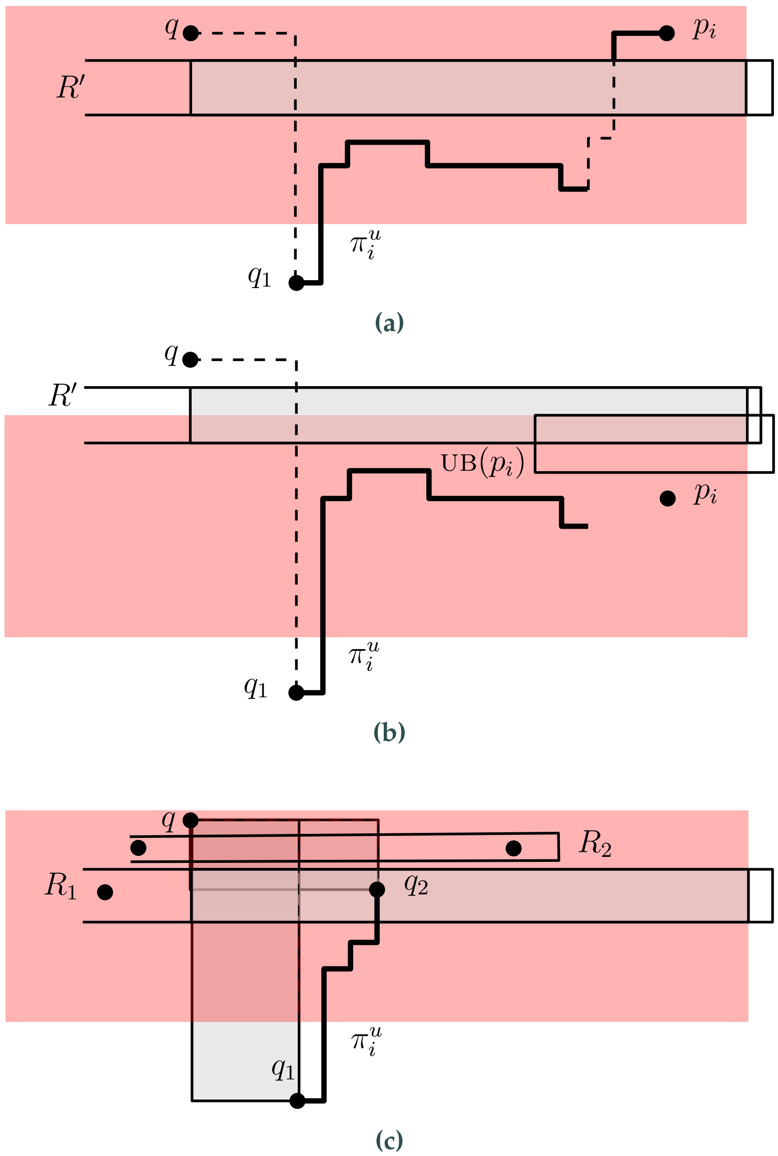


Figure 5.6: (a): R' cannot be empty if p_i is above R', (b): If R' does not contain a point of P between $x(q)$ and $x_+(R)$, then $y(q) > y_+(\textsc{Strip}_i)$, (c): Since qq_1 and qq_2 are Delaunay, then R_2 pierces $v(qq_1)$.

We show that there is a valid Delaunay edge between q and a point q' on π_i^u. By assumption, q lies above π_i^u. Let q_0 and q_1 denote, respectively, the left- and right-neighbors of q on π_i^u.

Since q is minimal in P_i', qq_1 is Delaunay. By Proposition 5.5, it follows that $y(q) > y(q_0)$ and $y(q) > y(q_1)$. Since q is not left-adjacent to q_1 on π_i^u, it implies qq_1 is not valid.

Since qq_1 is Delaunay but not valid, by Lemma 5.3, $v(qq_1)$ pierces a rectangle. Let $\mathcal{R}'$ denote the set of rectangles pierced by qq_1. Suppose there exists an $R' \in \mathcal{R}'$ such that $R'[x(q_1), \min\{x(p), x_+(R')\}] \cap P = \emptyset$. Then, we call R' a *bad* rectangle. Otherwise, we say that R' is *good*. Now we split the proof into two cases depending on whether

$\mathcal{R}'$ contains a bad rectangle or not. In the two cases below, we use Proposition 5.3 that π_i^u is x-monotone. See Fig. 5.6 for illustration.

Case 1. $\mathcal{R}'$ contains a bad rectangle. Let $R' \in \mathcal{R}'$ be a bad rectangle. First, observe that $x_+(R') > x(p)$ as otherwise, R' is not pierced by $v(qq_1)$. Now, suppose $y(p_i) > y_+(R')$, where p_i is the rightmost point in $\textsc{Strip}_i$. We have that $x(q_1) < x(p_i)$, both p_i and q_1 lie on π_i^u and, π_i^u is x-monotone. But, this implies π_i^u pierces R'. But this is impossible as by construction π_i^u consists of valid Delaunay edges. Hence, since R' is bad, we can assume that $y(p_i) < y_-(R')$. But the definition of the upper barrier implies that $y_+(R') \geq y_+(\textsc{Ub}(p_i))$. Since $v(qq_1)$ pierces R', it implies $y(q) > y_+(R')$, and hence $y(q) > y_+(\textsc{Ub}(p_i))$. But, this contradicts the assumption that $q \in \textsc{Strip}_i$, since by Proposition 5.1, $y_+(\textsc{Ub}(p_i)) \geq y_+(\textsc{Strip}_i)$ and hence $y(q) > y_+(\textsc{Strip}_i)$.

Case 2. All rectangles in $\mathcal{R}'$ are good. Let $R_1 = \arg\max\{y_-(R') : R' \in \mathcal{R}'\}$. Let q_2 be the leftmost point in R_1 s.t. $x(q_2) > x(q_1)$. Since q_2 lies to the right, and below q, $q_2 \prec q$. Since q is a minimal element in P_i', it implies that q_2 lies on or below π_i^u. We claim that q_2 cannot lie below π_i^u. Suppose it did. Let q_2' be the left-neighbor of q_2 on π_i^u. Then, $x(q_2') < x(q_2)$. Since q_2 is the leftmost point of R_1 to the right of $v(qq_1)$, then either $y(q_2') < y_-(R_1)$, or $y(q_2') > y_+(R_1)$. However, in either case, we obtain that π_i^u must cross R_1 between q_2 and q_1, which implies that π_i^u pierces R_1, as π_i^u is x-monotone, and the edges in π_i^u are of the form $\{\urcorner, \lrcorner\}$, a contradiction.

Since q is minimal, qq_2 is Delaunay. By Lemma 5.3, the only reason qq_2 is not valid is that $v(qq_2)$ pierces a rectangle. But, any such rectangle R_2 is also pierced by $v(qq_1)$, as qq_2 is Delaunay. But, this implies $y_-(R_2) > y_-(R_1)$, contradicting the choice of R_1. Therefore, qq_2 is a valid Delaunay edge. Now, the only reason that q is not the left-adjacent point of q_2 on π_i^u then, is that q_2', the left-adjacent point of q_2 on π_i^u lies in $\textsc{Slab}(R, p)$, but above q, i.e., $y(q_2') > y(q)$, as the construction of π_i^u dictates that we choose the left-adjacent point with highest y-coordinate that lies in $\textsc{Slab}(R, p)$. Showing that this leads to a contradiction completes the proof.

So suppose $y(q_2') > y(q)$, then $y(q_2') > y_+(R_1)$. Further, by Proposition 5.2, $x(q_2') > x(q)$. Again this implies the x-monotone curve π_i^u cannot contain both q_2' and q_1 without piercing R_1. Hence, $y(q_2') < y_-(R_1) < y(q)$, but this contradicts the choice of q_2' as the left-adjacent point of q_2 on π_i^u since qq_2 is a valid Delaunay edge with $y(q_2') < y(q_2) < y(q)$. ■

The key property of a corridor is that if the upper or lower path of a corridor crosses a rectangle R', then there must be a point of $R' \cap P$ that lies on that path of the corridor. Using this, we can show that any point in the strip has an adjacent point to its right in G that lies in the corridor. This implies that every point in a strip has a path to the rightmost point in the strip that lies entirely in its corresponding corridor.

Lemma 5.5. For each $q \in \text{CORRIDOR}_i$, there is a path $\pi(q, p_i)$ between q and p_i that lies in CORRIDOR_i, where $p_i \in P_i$ is the rightmost point in STRIP_i.

Proof. If q lies on the upper path π_i^u or the lower path π_i^ℓ defining CORRIDOR_i, the lemma is immediate. So we can assume by Proposition 5.4 that q lies below π_i^u, and above π_i^ℓ. Suppose the lemma is false. Let q be the rightmost point of CORRIDOR_i that does not have a path to p_i lying in CORRIDOR_i. To arrive at a contradiction, it is enough to show that q is adjacent to a point $q' \in \text{CORRIDOR}_i$ that lies to the right of q.

Starting from ℓ_q, the vertical line through q, sweep to the right until the first point r that lies on both π_i^u and π_i^ℓ. Such a point exists since p_i lies on both π_i^u and π_i^ℓ. Let Q_i' denote the set of points in CORRIDOR_i whose x-coordinates lie between $x(q)$ and $x(r)$. This set is non-empty as it contains q and r. Hence, either $Q_i^+ = \{q' \in Q_i' : y(q') > y(q)\} \neq \emptyset$, or $Q_i^- = \{q' \in Q_i' : y(q') < y(q)\} \neq \emptyset$. If both are non-empty, let Q_i denote the set that contains a point with smallest x-coordinate. Otherwise, we let Q_i denote the unique non-empty set. Assume $Q_i = Q_i^+$. An analgous argument holds when $Q_i = Q_i^-$.

Define a partial order on Q_i, where for $a, b \in Q_i$, $a \prec b \Leftrightarrow x(a) < x(b)$ and $y(a) < y(b)$. Let $Q_i^{\min} = (q_1, \ldots, q_t)$ denote the sequence of minimal elements of Q_i ordered linearly such that $y(q_k) > y(q_j)$ for $k < j$. It follows that $x(q_k) < x(q_j)$ for $k < j$. Observe that qq_i is Delaunay for $i = 1, \ldots, t$ by the minimality of q_i. Our goal is to show that qq_i is a valid Delaunay edge for some $i \in \{1, \ldots, t\}$. We start with the following claim that $v(qq_t)$ and $h(qq_1)$ do not pierce a rectangle in $\mathcal{R}$, or cross an edge of G.

Claim 5.1. $h(qq_1)$ does not pierce a rectangle in $\mathcal{R}$ or cross an edge of G, and $v(qq_t)$ does not pierce a rectangle in $\mathcal{R}$ or cross an edge of G.

Proof. Suppose $h(qq_1)$ pierced a rectangle R'. Since π_i^u consists of valid Delaunay edges, and the choice of Q_i, R' contains a point a that lies in CORRIDOR_i. Since $h(qq_1)$ pierces R', $x(q) < x(a) < x(q_1)$. If $y(a) < y(q_1)$, then it contradicts the fact that q_1 is minimal, and if $y(a) > y(q_1)$, it contradicts the fact that q_1 is the minimal element with highest y-coordinate. A similar argument shows that $h(qq_1)$ does not cross an edge of G.

If $v(qq_t)$ pierced a rectangle $R' \in \mathcal{R}$, then R' has a point to the right of $v(qq_t)$. Further, by Proposition 5.3, π_i^u and π_i^ℓ are x-monotone paths and by construction, they consist of valid Delaunay edges meeting at r. If $r = q_t$ and $v(qq_t)$ pierced a rectangle, since qq_t is Delaunay, and the edges are of type $\{\urcorner, \lrcorner\}$, it implies that the left-adjacent point of r on π_i^u or π_i^ℓ is not a valid Delaunay edge. Hence, we can assume $q_t \neq r$. Again, since the edges of π_i^u and π_i^ℓ are valid Delaunay edges, it implies that R' has a point a s.t. $x(q_t) < x(a) \leq x(r)$, and a lies in $\textsc{Corridor}_i$. Since $v(qq_t)$ pierces R', $y(a) < y(q_t)$. But this contradicts the fact that q_t is the point in $Q_i^{\min}$ with the smallest y-coordinate. Therefore, $v(qq_t)$ can not pierce any rectangle. Since qq_t is Delaunay, by Proposition 5.6, $v(qq_t)$ does not cross an edge of G. This proves the claim.

We now define a point x_1 on the x-axis and a point y_1 on the y-axis as follows:

$$x_1 = \min\{\{x_+(R') : R' \text{ pierced by } h(qq_t)\}, \{v(e) : e \text{ crosses } h(qq_t)\}\}$$
$$y_1 = \min\{\{y_+(R') : R' \text{ pierced by } v(qq_1)\}, \{h(e) : e \text{ crosses } v(qq_1)\}\}$$

By Claim 5.1 and the assumption that qq_t and qq_1 are not valid edges, it follows that x_1 and y_1 exist. We argue when x_1 and y_1 correspond to $x_+(R')$ and $y_+(R'')$, respectively for rectangles $R', R'' \in \mathcal{R}$. If they were instead defined by the vertical/horizontal side of edges, the arguments are similar.

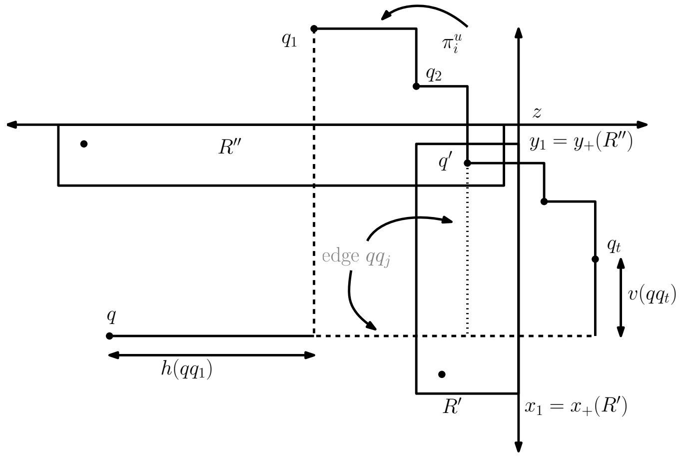


Figure 5.7: The edge qq' is a valid Delaunay edge.

Observe that $x_-(R'') < x(q)$, while $x_-(R') > x(q)$, and $y_-(R'') > y(q)$ and $y_-(R') < y(q)$. Now, from the fact that the rectangles are non-piercing, it implies that either $x_+(R'') < x_1$ or $y_+(R') < y_1$. Suppose without loss of generality, the former is true. Since R' is pierced by $h(qq_t)$ and π_i^u consists of valid Delaunay edges, there are points in R' that lie in $\textsc{Corridor}_i$, and these points lie below y_1.

Let z denote the intersection of the vertical line through x_1 and the hoizontal line through y_1. By the argument above, the rectangle with diagonal qz contains points of P, and hence a point $q' \in Q_i^{\min}$. We claim that qq' is a valid Delaunay edge. To see this, note that $h(qq')$ does not pierce a rectangle in $\mathcal{R}$ as such a rectangle contradicts the definition of x_1. If $v(qq')$ pierced a rectangle, such a rectangle $\tilde{R}$ must have $y_+(\tilde{R}) < y_1$, as qq' is Delaunay. This contradicts the choice of y_1. Therefore, qq' is a valid Delaunay edge. ■

The lemma below follows the description in the proof strategy at the start of this section.

Lemma 5.6. For a rectangle R and point $p \in R$, after Algorithm 1 has processed point p, the points in $\textsc{Slab}(R, p)$ induce a connected subgraph, all of whose edges lie in $\textsc{Slab}(R, p)$.

Proof. Let $G[\textsc{Slab}(R, p)]$ denote the induced subgraph of G on the points in $\textsc{Slab}(R, p)$. By Condition (ii) of Lemma 5.1, since $\textsc{Slab}(R, p) \subseteq \cup_{i=-t}^{k} \textsc{Strip}_i$, each point in $P \cap \textsc{Slab}(R, p)$ is contained in $\cup_{i=-t}^{k} \textsc{Strip}_i$. If the statement of the lemma does not hold, consider an extremal strip, i.e., the smallest positive index, or largest negative index of a strip such that it contains a point q that does not lie in the connected component of $G[\textsc{Slab}(R, p)]$ containing p. Assume without loss of generality that $i \geq 0$. An analogous argument holds if $i < 0$. By Lemma 5.4, $q \in \textsc{Corridor}_i$, and by Lemma 5.5, q has a path π_1 to p_i, the rightmost point in $\textsc{Corridor}_i$ that lies entirely in $\textsc{Corridor}_i$. By Condition (ii) of Lemma 5.1, there is a point $q' \in \textsc{Strip}_i \cap \textsc{Strip}_{i-1} \cap P$. By Lemma 5.4, $q' \in \textsc{Corridor}_i$, and by Lemma 5.5, there is a path π_2 between q' and p_i. Since $q' \in \textsc{Strip}_{i-1}$, q' lies in the same connected component as p in $G[\textsc{Slab}(R, p)]$, and hence there is a path π' from q' to p in $G[\textsc{Slab}(R, p)]$. Concatenating π_1, π_2 and π' we obtain a path π from q to p that lies in $\textsc{Slab}(R, p)$. ■

We now argue that if p is the rightmost point in a rectangle R, then $\textsc{Piece}(R, p)$ consists of a single slab.

Lemma 5.7. If p is the last point in R according to the x-coordinates of the points, then $\textsc{Piece}(R, p)$ consists of a single slab.

Proof. Assume for the sake of contradiction that $\textsc{Upb}(R, p)$ exists. By definition of $\textsc{Upb}(R, p)$, there are two points $a, b \in \textsc{Upb}(R, p)$, such that $x(a) < x_-(R) < x(p) < x_+(R) < x(b)$, as p is the last point in R. But this implies $\textsc{Upb}(R, p)$ is pierced by R, a contradiction. Therefore, $\textsc{Upb}(R, p)$ does not exist. Similarly, $\textsc{Lpb}(R, p)$ does not exist, and hence $\textsc{Piece}(R, p)$ consists of a single slab. ■

Theorem 5.2. If $(P, \mathcal{R})$ is a hypergraph defined by a point set P and a set $\mathcal{R}$ of axis-aligned non-piercing rectangles in the plane, Algorithm 1 constructs a planar support for $(P, \mathcal{R})$.

Proof. By construction, the edges of the graph G constructed by Algorithm 1 are valid Delaunay edges of type $\{\urcorner, \lrcorner\}$. To obtain a plane embedding, we replace each edge $e = \{p, q\}$ by the diagonal of the rectangle $R(pq)$ joining p and q. We call these the *diagonal edges*. It is clear that no diagonal edge pierces a rectangle. If two diagonal edges cross, then it is easy to check that either the corresponding edges cross, or they are not Delaunay. For a rectangle $R \in \mathcal{R}$, let p be the last point in R. Lemma 5.7 implies that there is only one slab, namely R, and Lemma 5.6 implies $\textsc{Slab}(R, p)$ is connected. Since R was arbitrary, this implies Algorithm 1 constructs a support. ■

§ 5.3.2 Implementation

In this section, we show that Algorithm 1 can be implemented to run in $O(n \log^2 n + (m+n)\log m)$ time with appropriate data structures, where $|\mathcal{R}| = m$, and $|P| = n$. At any point in time, our data structure maintains a subset of points that lie to the left of the sweep line ℓ. It also maintains for each rectangle R intersecting ℓ, the interval $[y_-(R), y_+R]$ corresponding to R. When the sweep line arrives at the left side of a rectangle, the corresponding interval is inserted into the data structure. The interval is removed from the data structure when the sweep line arrives at the right side of the rectangle. Similarly, whenever we sweep over a point p, we insert it into the data structure. In addition, we do the following when the sweep line arrives at a point p:

(S-1) Find the upper and lower barriers at p.

(S-2) Query the data structure to find the set Q of points q which $i)$ lie to the left of p and between the upper and lower barriers at p (orthogonal range query) so that $ii)$ qp is a Delaunay edge.

(S-3) We add the edge qp for every $q \in Q$ to our planar support. For each edge we add, we remove the points in the data structure that are *occluded* by the edges. These are the points whose y-coordinates lie in the range corresponding to the vertical side of the L-shape for qp.

Our data structure is implemented by combining three different existing data structures. For step (S-1), we use a balanced binary search tree $\mathcal{T}_1^u$ augmented so that it can answer range minima or maxima queries. For any rectangle R intersecting the sweep line ℓ, let (y_1, y_2) denote the interval corresponding to the projection of R on the y-axis. $\mathcal{T}_1^u$ stores the key-value pair (y_1, y_2) with y_1 as the key and y_2 as the value. To find the upper barrier at a point $p = (x, y)$ we need to find the smallest value associated with keys that are at least x. If we augment a standard balanced binary search tree so that at each node we also maintain the smallest value associated with the keys in the subtree rooted at that node, such a query takes $O(\log m)$ time. An analogous search tree $\mathcal{T}_1^b$ is used to find the lower barrier at any point.

To implement step (S-2), we use a dynamic data structure $\mathcal{T}_2^b$ due to Brodal [BT11] which maintains a subset of the points to the left of ℓ and can report points in any query rectangle Q that are not dominated by any of the other points in time $O(\log^2 n + k)$ where k is the number of reported points. We say that a point u is dominated by a point v if both x and y coordinates of u are smaller than those of v. The data structure also supports insertions or deletions of points in $O(\log^2 n)$ time. When the sweep line arrives at a point p, we can use $\mathcal{T}_2^b$ to find all points q that lie to the left of p and below p so that the edge qp is a Delaunay edge (as qp of shape $\lrcorner$ is Delaunay iff there is no other point in the range below and to the left of p that dominates q). An analogous data structure $\mathcal{T}_2^u$ is used to find the points q which lie above and to the left of p so that qp (of shape $\urcorner$) is Delaunay.

To implement step (S-3), we use a dynamic 1D range search data structure $\mathcal{T}_3$ which also stores a subset of the points to the left of ℓ, supports insertions and deletions in $O(\log n)$ time and can report in $O(\log n + k)$ time the subset of stored points that lie in a given range of y-coordinates (corresponding the vertical side of each added edge), where k is the number of points reported. The points identified are removed from $\mathcal{T}_2^u, \mathcal{T}_2^b$ and $\mathcal{T}_3$.

By the correctness of Algorithm 1 proved in Section 5.3.1, at any point in time, the current graph is a support for the set of rectangles that lie completely to the left of the sweep line. Thus, if the sweep line ℓ is currently at a point p and q is a point to the left of ℓ, the only rectangles that qp may discretely pierce are those that intersect ℓ. A simple but important observation is that if qp is Delaunay then qp pierces a rectangle iff the vertical portion of L-shape forming the edge pq pierces the rectangle. To see this, note that the horizontal portion of the L-shape cannot pierce any rectangle since such a rectangle would not intersect ℓ. The L-shape also cannot (discretely) pierce a rectangle containing the corner of the L-shape since then the edge qp would not be a Delaunay edge. Thus, in order to avoid edges that pierce other rectangles, it suffices to restrict q to lie between the upper and lower barriers at p. Thus, step (S-1) above ensures that edges found in step (S-2) don't pierce any of the rectangles. Similarly, step (S-3) ensures that the edges we add in step (S-2) don't intersect previously added edges.

The overall time taken by the data structures used by step (S-1) is $O((m + n) \log m)$ since it takes $O(\log m)$ time to insert or delete the key-value pair corresponding to any of the m rectangles, and it takes $O(\log m)$ time to query the data structure for the upper and lower barriers at any of the n points. The overall time taken by the data structure in Step 2 is $O(n \log^2 n)$ since there are at most $O(n)$ insert, delete, and query operations, and the total number of points reported in all the queries together is $O(n)$. The overall time taken by the data structure in step (S-3) is $O(n \log n)$ we only add $O(n)$ edges in the algorithm, and the query corresponding to each edge takes $O(\log n)$ time. Each of the reported points is removed from the data structure, but since each point is removed only once, the overall time for such removals is also $O(n \log n)$. The overall running time of our algorithm is therefore $O(n \log^2 n + (m + n) \log m)$.

§ 5.3.3 Lower bound on the running time

We show that any algorithm to compute a planar support for axis-parallel rectangles and points in the plane, has running time $\Omega(n \log n)$. This can be shown via reduction from *sorting*. Given an instance of sorting an array A of numbers $1, 2, \ldots n$. Let $P = \{(i, 0) : 1 \leq i \leq n\}$ be the set of n points in $\mathbb{R}^2$. For $i = 1, 2, \ldots, n-1$, let R_i be the rectangle defined by $[i - \epsilon, i + 1 + \epsilon] \times [-\delta_i, \delta_i]$ for some $\epsilon > 0$, and $0 < \delta_1 < \delta_2 < \ldots < \delta_n$. Let $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$. Clearly, $\mathcal{R}$ is a set of axis-aligned non-piercing rectangles in $\mathbb{R}^2$. Since $|A| = n$, the construction of $\mathcal{R} \cup P$ takes $O(n)$ time. Any support for the hypergraph $(P, \mathbb{R}^2)$ contains a path $1, 2, \ldots, n$. This also gives the sorting of the elements in A in increasing order. If $T(n)$ is the time taken to compute the support, then one can sort the elements of A in time $O(n) + T(n)$. Hence, $T(n) = \Omega(n \log n)$.

§ 5.4 General Families of Rectangles

In this section, we construct an embedded (not-necessarily planar) support graph G for an arbitrary family $\mathcal{R}$ of non-piercing rectangles using the results in the previous section. If $\mathcal{R}$ can be partitioned into t families of non-piercing rectangles, then G is a union of t plane graphs.

We show that there is a polynomial-time algorithm to partition a family $\mathcal{R}$ of axis-parallel rectangles into a minimum number of parts, each of which is a collection of non-piercing rectangles.

To that end, we define the following:

Definition 5.6 (Piercing graph)**.** For a set $\mathcal{R}$ of axis-parallel rectangles, the *piercing graph* is a graph $H = (\mathcal{R}, E)$ on the set $\mathcal{R}$ where for each $R_1, R_2 \in \mathcal{R}$, $\{R_1, R_2\} \in E$ if $R_1 \setminus R_2$ (and hence $R_2 \setminus R_1$) is disconnected.

We claim that the piercing graph H is a *comparability graph*. A graph on vertex set

V is a comparability graph if H can be obtained from the transitive closure of a partial order on V by forgetting directions.

Theorem 5.3. For a set $\mathcal{R}$ of axis-parallel rectangles in the plane, the piercing graph H is a comparability graph.

Proof. For a rectangle $R \in \mathcal{R}$, let $x(R)$ and $y(R)$ denote respectively, the projections of R on the x and y axes. Two rectangles R_1 and R_2 pierce if and only if $x(R_1) \subseteq x(R_2)$, and $y(R_2) \subseteq y(R_1)$ or vice versa. Consider the containment order $\prec_x$, where $R_1 \prec_x R_2$ if and only if $x(R_1) \subseteq x(R_2)$. Similarly, consider the containment order $\prec_y$, where $R_1 \prec_y R_2$ if and only if $y(R_2) \subseteq y(R_1)$ (note that the order is reversed). That is, the partial order on the rectangles induced by their y-projection is the reverse of the containment order. Now, R_1 and R_2 are adjacent in G if and only if $R_1 \preceq_x R_2$ and $R_1 \preceq_y R_2$. Since the intersection of two partial orders is a partial order, we obtain a partial order $\prec_P$ where $R_1 \prec_P R_2$ iff R_1 and R_2 pierce. We obtain H from $\prec_P$ by completing the transitive closure of $\prec_P$ and forgetting the directions of the edges. ■

Comparability graphs are perfect, they can be colored optimally in polynomial time [Gol80a; Gol80b]. In our context, this means that we assign colors to the rectangles so that piercing rectangles receive distinct colors. By Theorems 5.2 and 5.3, we get the following result.

Theorem 5.4. Let $\mathcal{R}$ be a collection of axis-parallel rectangles, and P be a set of points such that the piercing graph has chromatic number k. Then, there is a collection of k planar graphs whose union is a support for the hypergraph $(P, \mathcal{R})$.

Proof. The piercing graph of $\mathcal{R}$ is a comparability graph by Theorem 5.3. Since we can compute the chromatic number and a coloring with the smallest number of colors in polynomial time for such graph classes [Gol80b], we can partition $\mathcal{R}$ into k color classes $\mathcal{R}_1, \ldots, \mathcal{R}_k$ such that each color class consists of a set of non-piercing rectangles. By Theorem 5.2, for each color class $\mathcal{R}_i$ and points contained in the rectangles in $\mathcal{R}_i$, we can construct a planar support. It can be easily verified that the union of these support graphs yields a support for $(P, \mathcal{R})$ as a union of k planar graphs. ■

§ 5.5 Concluding Remarks and Open Questions

We presented a fast algorithm to compute a plane support graph for non-piercing axis-parallel rectangles that runs in $O(n \log^2 n + (n + m) \log m)$ with n being the number of points and m, the number of rectangles. This significantly improved the running time of the previous work by Raman and Ray [RR20], who gave a $O(m^2(\min\{m^3, mn\} + n))$ time algorithm to compute planar support for non-piercing regions. We also gave a lower bound of $\Omega(n \log n)$ for the running time to compute a planar support in our context.

Although the algorithm itself is simple, the fast implementation relies on sophisticated orthogonal range searching data structures of Brodal et al., [BT11], and the proof of correctness itself turned out to be surprisingly complex.

▷ Limitations.

As mentioned above, we gave lower and upper bounds on the running time to compute a planar support. The two bounds are close to each other, but we could not eliminate the gap. Further, the algorithm presented in this chapter is only for the primal case. It seems plausible that our machinery could be extended to the dual and intersection settings, though we were not able to achieve this here. With this discussion, we propose the following open problems.

> **Open Problem 9.** Is it possible to improve the running time of Algorithm 1 to $(n + m) \log(n + m)$.

Motivated by the work of Raman and Ray [RR20], who obtained polynomial time algorithms to construct *dual* and *intersection* supports for non-piercing regions. In our context, these questions translate into the following:

> **Open Problem 10.** Given a set of points P and axis-parallel non-piercing rectangles $\mathcal{R}$ in the plane, obtain a fast algorithm to construct a plane support graph $Q^* = (\mathcal{R}, E)$ on $\mathcal{R}$ such that for each point $p \in P$, the rectangles containing p induce a connected subgraph in Q^*.

> **Open Problem 11.** Given two sets $\mathcal{R}$, $\mathcal{S}$, each of which is a collection of non-piercing rectangles, obtain a fast algorithm to construct a plane support graph $\tilde{Q}$ on $\mathcal{R}$, i.e., a graph $\tilde{Q} = (\mathcal{R}, F)$ such that for each $S \in \mathcal{S}$, the induced graph of $\tilde{Q}$ on $\{R \in \mathcal{R} : R \cap S \neq \emptyset\}$ is connected.

(B) Applications

Chapter 6

Applications of Supports

Abstract

In this chapter, we describe applications of support and its related weaker notions to the packing and covering problems. We show PTAS for these problems when the underlying hypergraphs arise from a surface of bounded genus, i.e., they are defined either by subgraphs of a bounded genus graph or by geometric regions on such surfaces. We also show some hardness results and results on hypergraph coloring. The majority of the content in this chapter is based on the following paper.
"**On Supports for graphs of bounded genus**"

Table of Contents

§ 6.1 Preliminaries

In this section, we introduce basic definitions that we require throughout this chapter. Most of the notions we define in this section are related to packing and covering problems on hypergraphs.

We start with definitions of the problems we consider. The input for the problems below is a set X along with a collection of subsets $\mathcal{S}$ of X.

Definition 6.1 (Set Packing)**.** Pick a *largest* sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ such that $|\{S \in \mathcal{S}' : S \ni x\}| \leq 1$ for all $x \in X$.

Definition 6.2 (Point Packing)**.** Pick a *largest* set $Y \subseteq X$ such that for all $S \in \mathcal{S}$, $|Y \cap S| \leq 1$.

Definition 6.3 (Set Cover)**.** Assuming $\cup_{S \in \mathcal{S}} S = X$, select a *smallest* sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ such that $\cup_{S \in \mathcal{S}'} S = X$.

Definition 6.4 (Hitting Set)**.** Select a *smallest* cardinality set $Y \subseteq X$ such that $Y \cap S \neq \emptyset, \forall S \in \mathcal{S}$.

We also define a generalized version of packing and covering problems. Let X be a set, and let $\mathcal{S}$ and $\mathcal{T}$ be two collections of subsets of X. This defines an *intersection hypergraph* $(\mathcal{S}, \{\mathcal{S}_T\}_{T \in \mathcal{T}})$, where for $T \in \mathcal{T}$, $\mathcal{S}_T = \{S \in \mathcal{S} : S \cap T \neq \emptyset\}$.

Definition 6.5 (Generalized Packing)**.** The Generalized Packing problem is the Set/Point Packing problem on the intersection hypergraph $(\mathcal{S}, \{\mathcal{S}_T\}_{T \in \mathcal{T}})$.

Definition 6.6 (Generalized Covering)**.** The Generalized Covering problem is the Set Cover/Hitting Set problem on the intersection hypergraph $(\mathcal{S}, \{\mathcal{S}_T\}_{T \in \mathcal{T}})$.

Definition 6.7 (Generalized Capacitated Packing)**.** Given an intersection hypergraph $(\mathcal{S}, \mathcal{T})$ and a capacity function $\mathrm{cap} : \mathcal{T} \to \mathbb{N}$, the goal is to find a largest sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ such that for each $T \in \mathcal{T}$, $|\mathcal{S}'_T| \leq \mathrm{cap}(T)$, where $\mathcal{S}'_T = \{S \in \mathcal{S}' : S \cap T \neq \emptyset\}$.

In the generalized version of the problems, taking either the set $\mathcal{S}$, or the set $\mathcal{T}$ as singleton sets of X, we obtain the Set Cover/Set Packing, or Point Packing/Hitting Set problems. Hence, any result obtained for the generalized version of the problems applies immediately to the classic versions of these problems.

We also study some graph problems on an intersection graph defined by a hypergraph. Let $(X, \mathcal{S})$ be a hypergraph. The *intersection graph* defined by $\mathcal{S}$ is the graph $I(\mathcal{S}) = (\mathcal{S}, F)$, where $\{S, S'\} \in F \Leftrightarrow S \cap S' \neq \emptyset$.

Definition 6.8 (Dominating Set)**.** Let $(X, \mathcal{S})$ be a hypergraph. For the intersection graph $I(\mathcal{S})$, a set $\mathcal{D} \subseteq \mathcal{S}$ is a Dominating Set if for each $S \in \mathcal{S}$, either $S \in \mathcal{D}$ or $S \cap S' \neq \emptyset$ for some $S' \in \mathcal{D}$. The goal is to find a dominating set of *minimum* cardinality.

Definition 6.9 (Independent Set)**.** Let $(X, \mathcal{S})$ be a hypergraph. For the intersection graph $I(\mathcal{S})$, a set $\mathcal{S}' \subseteq \mathcal{S}$ is an Independent Set if the sets in $\mathcal{S}'$ are pairwise non-adjacent. The goal is to find an independent set of *maximum* cardinality.

Definition 6.10 (Vertex Cover)**.** Let $(X, \mathcal{S})$ be a hypergraph. For the intersection graph $I(\mathcal{S})$, a set $\mathcal{C} \subseteq \mathcal{S}$ is a Vertex Cover if for each edge in $I(\mathcal{S})$ at least one of its end-points is in $\mathcal{C}$. The goal is to find a vertex cover of *minimum* cardinality.

Our next definition is about the separator of a graph that plays a crucial role in obtaining better approximation algorithms for several NP-hard optimization problems.

Definition 6.11 (Sub-linear sized balanced separator)**.** For a graph $G = (V, E)$ a set $S \subsetneq V$ is a *separator* if $G \setminus S$ consists of two disjoint components A and B. A separator is said to have *sub-linear size* if $|S| = O(n^\delta)$ for a constant $\delta < 1$. A separator is said to be balanced if $|A|, |B| \leq \alpha|V|$ for some constant $\alpha < 1$.

We also need the notion of a *weighted separator*. Let $w : V \rightarrow \mathbb{R}_{\geq 0}$ be a weight function on the vertices. A separator S is said to be *balanced* if $w(A), w(B) \leq \alpha w(V)$, where $w(X) = \sum_{v \in X} w(v)$. For the separator S, its size is still its cardinality. The key property of graphs of bounded genus that we use is that they have sublinear-sized balanced separators.

Note 6.1.1. In the rest of this chapter, since we only deal with sub-linear balanced separators, we will just use the term separator to mean a sublinear-sized, balanced separator.

We will make use of the following result due to Gilbert et al. [GHT84].

Theorem 6.1 ([GHT84]). Let $G = (V, E)$ be an n-vertex graph of genus g. Let $w : V \to \mathbb{R}_{\geq 0}$ be a weight function. Then, G has a balanced separator of size $O(\sqrt{gn})$ such that $w(A), w(B) \leq \frac{1}{2}w(V)$.

Definition 6.12 (PTAS). Let Π be an optimization problem. An algorithm $\mathcal{A}$ is said to be a PTAS (polynomial-time approximation scheme) for Π, if for every fixed $\epsilon > 0$, $\mathcal{A}$ produces a solution that is within $(1 \pm \epsilon)$ factor of being optimal (depending on whether Π is a minimization or a maximization problem).

In abstract set systems, the Point Packing problem is just the Set Packing problem on the dual set system $(\mathcal{S}, \{S_x\}_{x \in X})$, where $S_x = \{S \in \mathcal{S} : S \ni x\}$. Hence, algorithmic and hardness results that hold in the primal also hold in the dual. The Set Packing problem contains as a special case, the Independent Set problem on graphs, and is therefore hard to approximate beyond $n^{1-\epsilon}$ for any $\epsilon > 0$ unless NP=ZPP [Hå99].

As in the case of packing problems, in abstract set systems, the Hitting Set problem is just the Set Cover problem on the dual hypergraph. It is well known that Set Cover admits an $O(\log n)$-approximation [Chv79; Lov75], and that this is tight [Fei98] up to lower order terms.

More generally, in the abstract setting, the generalized variants are no different from the classic problems, but they have different flavors, especially in geometric settings. For example, there exists a QPTAS[1] for the problem of covering a set of points in the plane with disks [MRR15], but such a result is not known for the Hitting Set variant of the problem.

A support is meant to capture the structure of a hypergraph. Besides a support, weaker notions have been used in many applications. We now define two such notions, assuming that every hyperedge contains at least two vertices.

Definition 6.13 (Weak support). Let $(X, \mathcal{E})$ be a hypergraph. A *weak support* for $(X, \mathcal{E})$ is a graph $Q' = (X, F)$ such that $\forall E \in \mathcal{E}$ containing at least two vertices, $\exists$ an edge $\{u, v\} \in F$ for some $u, v \in E$.

[1] A QPTAS is an algorithm whose approximation guarantee is the same as a PTAS, but whose running time is allowed to be a quasi-polynomial in the input, i.e., of the form $2^{polylog(n)}$.

Definition 6.14 (Weak bipartite support)**.** Let $(X, \mathcal{E})$ be a hypergraph and $c : X \to \{\mathrm{b}, \boldsymbol{r}\}$ be a 2-coloring of X. A *weak bipartite support* for $(X, \mathcal{E})$ is a bipartite graph $Q'' = (X, F)$ with bipartition $\boldsymbol{r}(X)$ and $\mathrm{b}(X)$ such that $\forall E \in \mathcal{E}$ with $E \cap \mathrm{b}(X) \neq \emptyset$ and $E \cap \boldsymbol{r}(X) \neq \emptyset$, $\exists$ an edge $\{u, v\} \in F$ with $u \in E \cap \mathrm{b}(X)$ and $v \in E \cap \boldsymbol{r}(X)$.

It is easy to see that a support Q is also a weak bipartite support for any 2-coloring of the vertex set. Indeed, if a hyperedge induces a connected subgraph of Q, and contains a vertex from both $\mathrm{b}(X)$ and $\boldsymbol{r}(X)$, there must be an edge between some vertex in $\mathrm{b}(X)$ and some vertex in $\boldsymbol{r}(X)$ that are in the hyperedge. Further, a weak bipartite support implies a weak support. However, it is not difficult to construct instances where a sparse weak bipartite support exists, but a sparse support does not.

As we will see in Section 6.4, for the applications of Hypergraph Coloring, what we require is a weak support, while for many (though not all) of the packing and covering problems we consider, what we require is a weak bipartite support.

§ 6.2 Packing and Covering via Local Search

In combination with previous work, our results on the existence of a sparse support imply that for a general class of geometric hypergraphs in the plane, or on an oriented surface of bounded genus, a simple *local search framework* leads to PTAS for all the problems defined above. We describe an idea of the local search framework and how it leads to a PTAS for several optimization problems. We refer the reader to [Asc+13; CH12; MR10] for concrete algorithms and analysis under this framework.

§ 6.2.1 Local search framework

For a parameter $k \in \mathbb{N}$, the algorithms in the framework have the following form: We start with an arbitrary feasible solution. While there is a feasible solution of better value within a k*-neighborhood* of the current solution, replace the current solution with this better solution. Formally, we replace at most k elements in the current feasible solution with the elements not in the solution. In the case of maximization, we add at most k elements to the current solution and withdraw fewer elements from it so that the updated solution is still a feasible solution, and for the minimization case, we extract at most k elements and add fewer elements so that the solution remains feasible. When no such replacements are possible, the algorithm returns the current feasible solution. These algorithms guarantee an approximation factor of $O(1 \pm 1/k)$ and have running time $O(N^{1/k^2})$, where N is the input size. This basic framework is a basis for several PTAS for geometric packing and covering problems.

Below, we elaborate on the framework for maximization problems as introduced in [CH12]. A similar argument works for the minimization problems.

In the following, let $\mathcal{S}$ be a given collection of subsets of a set X, and Π be a maximization problem on the set system $(X, \mathcal{S})$, i.e., find a subset $\mathcal{S}' \subseteq \mathcal{S}$ of maximum size satisfying some given constraints.

Definition 6.15 (k-locally optimalilty)**.** A feasible solution $\mathcal{L} \subseteq \mathcal{S}$ is *k-locally optimal* if $\nexists$ a subset $\mathcal{K} \subseteq \mathcal{S}$ with $|\mathcal{K}| \leq k$ and a subset $\mathcal{B} \subseteq \mathcal{L}$ with $|\mathcal{B}| < |\mathcal{K}|$ such that $(\mathcal{L} \setminus \mathcal{B}) \cup \mathcal{K}$ is also a feasible solution.

Obtaining a k-locally optimal solution.
We start with an arbitrary feasible solution, say $\mathcal{L} = \emptyset$. If $\mathcal{L}$ is not k-locally optimal, there exists a feasible solution $\mathcal{K} \subseteq \mathcal{S} \setminus \mathcal{L}$ of size at most k such that $\mathcal{K}$ intersects $\mathcal{L}$ at fewer than $|\mathcal{K}|$ elements. If $\mathcal{L} \cap \mathcal{K} = \mathcal{B}$, we replace $\mathcal{L}$ with $(\mathcal{L} \setminus \mathcal{B}) \cup \mathcal{K}$. This improves the size of $\mathcal{L}$ by at least 1. We continue until no further improvements are possible and thus, the process terminates with a k-locally optimal solution.

Computation.
Let $|\mathcal{S}| = n$. Since at each step, our solution size increases by at least 1, the process terminates in $O(n)$ steps. Further, in each step, there are at most $\binom{n}{k}$ possible choices for $\mathcal{K}$, and for each choice, it takes $O(nk)$ time to obtain the subset $\mathcal{B} \subseteq \mathcal{L}$. Thus, a k-locally optimal solution $\mathcal{L}$ can be computed in $O(n^{k+3})$ time, which is polynomial for bounded k.

Analysis: how the local search implies a PTAS?
The key to proving that the local search framework yields a PTAS is to show the existence of a graph satisfying the *local search property*. Such a graph is called a *local search graph*. Below, we define this for both maximization and minimization problems.

Definition 6.16 (Local search property)**.** For a maximization or minimization problem, let $\mathcal{L}$ denote a solution returned by the local search framework and let $\mathcal{O}$ be an optimal solution. A graph $G = (\mathcal{L} \cup \mathcal{O}, E)$ is said to satisfy the *local search property* if it satisfies the following *local* and *global* properties.

1. **Local property:** Below $N(.)$ refers to the neighborhood in G.
   (a) (Maximization) For any $\mathcal{O}' \subseteq \mathcal{O}$, $(\mathcal{L} \cup \mathcal{O}') \setminus N(\mathcal{O}')$ is a feasible solution.
   (b) (Minimization) For any $\mathcal{L}' \subseteq \mathcal{L}$, $(\mathcal{L} \setminus \mathcal{L}') \cup N(\mathcal{L}')$ is a feasible solution.
2. **Global property:** G comes from a hereditary family of graphs that has sublinear separators.

The local property captures a subset of the local moves, and the global property is used to bound the approximation factor guaranteed by the algorithm. The argument is roughly as follows.

If $\mathcal{L}$ is a solution returned by a local search algorithm, and $\mathcal{O}$ is an optimal solution, the goal is to show that, $|\mathcal{L}| \geq (1 - 1/k) \cdot |\mathcal{O}|$. We can always assume that $\mathcal{L} \cap \mathcal{O} = \emptyset$. Indeed, let $\mathcal{L}' = \mathcal{L} \setminus (\mathcal{L} \cap \mathcal{O})$ and $\mathcal{O}' = \mathcal{O} \setminus (\mathcal{L} \cap \mathcal{O})$. If $|\mathcal{L}'| \leq (1 + 1/k) \cdot |\mathcal{O}'|$, then $|\mathcal{L}| \leq (1+1/k) \cdot |\mathcal{O}|$. Note that a similar argument also works in the minimization case. Therefore, we assume without loss of generality that $\mathcal{L}$ and $\mathcal{O}$ are disjoint.

Let $G = (\mathcal{L} \cup \mathcal{O}, E)$ be a local search graph. Then there is a sublinear-sized balanced separator in G. We recursively find such separators in each of the balanced parts until each component has size at most k. Note that this is possible since G comes from a hereditary family admitting such separators. Let S be the resulting separator in G such that $G \setminus S$ partitions into disjoint components with vertex sets $V_1, V_2, \ldots, V_{\lceil n/k \rceil}$ each of size at most k.

Now we show that in each of the components, the local search solution is nearly as good as any optimal solution. To that end, let $\mathcal{L}_i = \mathcal{L} \cap V_i$, $\mathcal{O}_i = \mathcal{O} \cap V_i$, and $S_i = S \cap N(V_i)$ for $i = 1, 2, \ldots, \lceil n/k \rceil$. Let $\ell_i = |\mathcal{L}_i|$, $o_i = |\mathcal{O}_i|$, and $s_i = |S_i|$. We show that $\ell_i + s_i \geq o_i$ for each i. If possible, let $\ell_i + s_i < o_i$ for some i. By the local property of G, $(\mathcal{L} \cup \mathcal{O}_i) \setminus N(\mathcal{O}_i)$ is a feasible solution to Π. It follows that $\mathcal{L}$ is not k-locally optimal since otherwise we would have replaced $\mathcal{L}_i \cup (\mathcal{L} \cap S)$ with $\mathcal{O}_i$ to get a better solution.

The rest of the argument to show the approximation factor follows using the small size of the separator S, which depends on the hereditary family of G, and we omit these calculations here. We refer the reader to [Asc+13; CH12; MR10] for further details.

To conclude this discussion, recall the properties of the local search graph (Definition 6.16 above), which plays the central role in a local search algorithm, implying the required PTAS. Earlier, the existence of a support was used to obtain PTAS in the planar setting based on the fact that the existence of such support implies the existence of an appropriate local search graph; see [RR20] for example. Our contribution is in generalizing this to graphs of bounded genus, primarily because of the existence of an appropriate support. Since by Theorem 6.1, a graph of bounded genus has a sub-linear sized balanced separator, in all the results below for bounded genus graphs, we only need to show the existence of a graph that satisfies the local property 1.

▷ Set/Point Packing, Set Cover, Hitting Set.

As a warm-up, consider the Set Packing problem for the cross-free system $(G, \mathcal{H})$ of genus g. We want to select a largest set $\mathcal{H}' \subseteq \mathcal{H}$ so that each $v \in V(G)$ is contained in at most one $H \in \mathcal{H}'$. Let $\mathcal{L}$ denote the solution returned by local search, and let $\mathcal{O}$ denote an optimal solution. As argued, assume $\mathcal{L} \cap \mathcal{O} = \emptyset$ for simplicity. The local property that the local search graph on $\mathcal{L} \cup \mathcal{O}$ must satisfy is that for any $\mathcal{O}' \subseteq \mathcal{O}$, $(\mathcal{L} \cup \mathcal{O}') \setminus N(\mathcal{O}')$ is a feasible solution to the Set Packing problem. This implies that for each $v \in V$, if it belongs to a subgraph in $\mathcal{L}$ and a subgraph in $\mathcal{O}$, then there is an edge between some $L \in \mathcal{L}_v$ and some $O \in \mathcal{O}_v$, i.e., a weak bipartite support for the dual hypergraph defined by the graph system $(G, \mathcal{L} \cup \mathcal{O})$. By Theorem 2.4, there is a support of genus at most g in this setting, and this implies a weak-bipartite support of genus at most g. Therefore, a dual support satisfies the conditions of a local search graph, and thus we have a PTAS for the Set Packing problem.

In a similar vein, for the Point Packing, Set Cover, or Hitting Set problems, a weak bipartite support of bounded genus yields the desired local search graph (see Table 6.1). For the Generalized Packing/Covering problems however, we require a support in order to construct the desired local search graphs.

▷ Generalized Capacitated Packing.

We start with results for graphs of bounded genus that follow from the existence of supports. We show a PTAS for the Generalized Capacitated Packing problem for a cross-free intersection system $(G, \mathcal{H}, \mathcal{K})$ of genus g. By Theorem 2.5, $(G, \mathcal{H}, \mathcal{K})$ has an intersection support of genus at most g. The existence of a PTAS follows the general framework in [RR22]. Let $\mathcal{L}$ and $\mathcal{O}$ be, respectively, the solution returned by the local search algorithm and an optimal solution. We can assume that $\mathcal{L} \cap \mathcal{O} = \emptyset$. Our goal is to construct a graph satisfying the local search property. We do this as follows: First, we show that there is a set with $O(|\mathcal{L} \cup \mathcal{O}|)$ subgraphs in $\mathcal{K}$ that *witness the intersection* of an $L \in \mathcal{L}$ and an $O \in \mathcal{O}$ with a $K \in \mathcal{K}$. This follows from the fact that there is an intersection support of bounded genus and the Clarkson-Shor technique [CS89; Sha03]. We next use these witness subgraphs to define a new intersection system and construct an intersection support for this system. Notably, in this new intersection system, the roles of $\mathcal{H}$ and $\mathcal{K}$ are reversed. Using the intersection support thus created, we are finally able to construct a graph on $\mathcal{L} \cup \mathcal{O}$ and show that the resulting graph satisfies the Local Search Properties required to show the existence of a PTAS.

Lemma 6.1. Let $(G, \mathcal{H}, \mathcal{K})$ be a cross-free intersection system of genus g such that for all $K \in \mathcal{K}$, $2 \leq |\mathcal{H}_K| \leq \Theta$ for some $\Theta \geq 2$. Then, there is a set $\mathcal{K}' \subseteq \mathcal{K}$ with $|\mathcal{K}'| = O(\Theta \cdot (g + |\mathcal{H}|))$ and such that for any pair $H, H' \in \mathcal{H}_K$ for some $K \in \mathcal{K}$, there is a $K' \in \mathcal{K}'$ such that $H, H' \in \mathcal{H}_{K'}$.

Proof. Let $\mathcal{K}' \subseteq \mathcal{K}$ be a minimal subset such that for any $H, H' \in \mathcal{H}_K$ for some $K \in \mathcal{K}$, there is a $K' \in \mathcal{K}'$ s.t. $H, H' \in \mathcal{H}_{K'}$. By minimality of $\mathcal{K}'$, for any $K' \in \mathcal{K}'$, there is a pair

$H, H' \in \mathcal{H}$ that appear together only in $\mathcal{H}_{K'}$.

By Theorem 2.5, there is an intersection support $\tilde{Q}$ of genus at most g for $(G, \mathcal{H}, \mathcal{K}')$. The construction in Theorem 2.5 also produces an embedding of $\tilde{Q}$ on the surface. We can assume without loss of generality that $\tilde{Q}$ is simple and is 2-cell embedded. Since $\tilde{Q}$ has genus at most g, by Euler's formula $|E(\tilde{Q})| \leq 3(g + |\mathcal{H}|)$. If a pair of subgraphs $H, H' \in \mathcal{H}_{K'}$ s.t. $|\mathcal{H}_{K'}| = 2$, then H and H' are adjacent in $\tilde{Q}$ since $\tilde{Q}$ is a support. Therefore, the number of $K' \in \mathcal{K}'$ s.t. $|\mathcal{H}_{K'}| = 2$ is at most $3(g + |\mathcal{H}|)$.

Let $\mathcal{H}'$ be a random subset of $\mathcal{H}$ where each $H \in \mathcal{H}$ is chosen independently with probability $p = 1/\Theta$. Consider a $K' \in \mathcal{K}'$. Since $|\mathcal{H}_{K'}| \leq \Theta$, the probability that $H, H' \in \mathcal{H}_{K'}$ are the only subgraphs of $\mathcal{H}_{K'}$ chosen in $\mathcal{H}'$ is at least $p^2(1-p)^{\Theta-2} \geq 1/(e\Theta)^2$. The expected number of subgraphs $K' \in \mathcal{K}'$ for which this happens is $|\mathcal{K}'|/(e\Theta)^2$. Since $\mathsf{E}[|\mathcal{H}'|] = |\mathcal{H}|/\Theta$, and there are at most $3(g + |\mathcal{H}|)$ pairs $H, H' \in \mathcal{H}'$ s.t. $\mathcal{H}_{K'} = \{H, H'\}$ for some $K' \in \mathcal{K}'$, by the minimality of $\mathcal{K}'$ it follows that $|\mathcal{K}'|/(e\Theta)^2 \leq 3(g + |\mathcal{H}|)/\Theta$. This implies $|\mathcal{K}'| \leq 3e^2\Theta \cdot (g + |\mathcal{H}|) = O(\Theta \cdot (g + |\mathcal{H}|))$. ■

Theorem 6.2. Let $(G, \mathcal{H}, \mathcal{K})$ be a cross-free intersection system of genus g for some constant g, and let cap : $\mathcal{K} \to \mathbb{N}$ be a capacity function. Let $\Delta > 0$ be an absolute constant. If $cap(K) \leq \Delta$ for all $K \in \mathcal{K}$, then the **Generalized Capacitated Packing** problem admits a PTAS via the local search framework.

Proof. Let $\mathcal{L} \subseteq \mathcal{H}$ be a solution returned by the local search algorithm and let $\mathcal{O} \subseteq \mathcal{H}$ be an optimal solution. Again, since we are interested in bounding $|\mathcal{L}|$ by $|\mathcal{O}|$, we remove the subgraphs in $\mathcal{L} \cap \mathcal{O}$ and replace the capacity of each $K \in \mathcal{K}$ by its *residual capacity*: $\mathrm{cap}'(K) = \mathrm{cap}(K) - |\{X : X \cap K \neq \emptyset \text{ and } X \in \mathcal{L} \cap \mathcal{O}\}|$. We also remove the subgraphs in $\mathcal{K}$ whose residual capacity is 0. Let $\mathcal{L}' = \mathcal{L} \setminus (\mathcal{L} \cap \mathcal{O})$ and $\mathcal{O}' = \mathcal{O} \setminus (\mathcal{L} \cap \mathcal{O})$. If we prove $|\mathcal{L}'| \leq (1+\epsilon)|\mathcal{O}'|$, then it implies that $|\mathcal{L}| \leq (1+\epsilon)|\mathcal{O}|$. Hence, we assume below that $\mathcal{L} \cap \mathcal{O} = \emptyset$, and each $K \in \mathcal{K}$ has positive capacity bounded above by Δ.

Consider the intersection system $(G, \mathcal{L} \cup \mathcal{O}, \mathcal{K})$. By Lemma 6.1, there is a collection $\mathcal{K}' \subseteq \mathcal{K}$ with $|\mathcal{K}'| = O(\Delta \cdot (g + |\mathcal{L} \cup \mathcal{O}|))$ s.t. for any $H, H' \in \mathcal{L} \cup \mathcal{O}$ that belong to $(\mathcal{L} \cup \mathcal{O})_K$ for some $K \in \mathcal{K}$, there is a $K' \in \mathcal{K}'$ s.t. $H, H' \in (\mathcal{L} \cup \mathcal{O})_{K'}$.

Now, consider the intersection system $(G, \mathcal{K}', \mathcal{L} \cup \mathcal{O})$. By Theorem 2.5, there is an intersection support $\tilde{Q}$ on $\mathcal{K}'$ of genus at most g. Abusing notation, we use K' to also denote the vertex in $\tilde{Q}$ corresponding to $K' \in \mathcal{K}'$. For each $H \in \mathcal{L} \cup \mathcal{O}$, we use H to also denote the induced subgraph of $\tilde{Q}$ defined by the vertices corresponding to subgraphs $K' \in \mathcal{K}'$ s.t. $K' \cap H \neq \emptyset$.

Let $\Xi = (\mathcal{L} \cup \mathcal{O}, E)$ denote the bipartite intersection graph of $\mathcal{L}$ and $\mathcal{O}$. That is, there is an edge between an $L \in \mathcal{L}$ and an $O \in \mathcal{O}$ iff there is a $K' \in \mathcal{K}'$ that intersects both L and O.

We claim that Ξ satisfies the local search condition for maximization: That is, for any $\mathcal{O}' \subseteq \mathcal{O}$, the modified solution $(\mathcal{L} \cup \mathcal{O}') \setminus N(\mathcal{O}')$ is feasible, where $N(\mathcal{O}') = \cup_{O \in \mathcal{O}'} N_{\Xi}(O)$ is the union of neighbors of each $O \in \mathcal{O}'$ in the graph Ξ. To see this, consider a $K \in \mathcal{K}$. If there is an $L \in \mathcal{L}_K$ and an $O \in \mathcal{O}_K$, then L and O are adjacent in Ξ. Since $\mathcal{L}$ and $\mathcal{O}$ are feasible, $|\mathcal{L}_K| \leq \text{cap}(K)$ and $|\mathcal{O}_K| \leq \text{cap}(K)$. Hence, we added at most $|\mathcal{O}_K|$ subgraphs for K, but removed all the subgraphs in $\mathcal{L}_K$. Therefore, $(\mathcal{L} \cup \mathcal{O}') \setminus N(\mathcal{O}')$ is feasible.

The graph Ξ may not have bounded genus. Nevertheless, we show that Ξ has a sub-linear separator. Each subgraph $H \in \mathcal{L} \cup \mathcal{O}$ assigns a weight of $1/|H|$ to the vertices of H in the intersection support $\tilde{Q}$. The weight of a vertex in $\tilde{Q}$ is the sum of the contributions it receives from the subgraphs containing it. Let $w(K)$ denote the weight of a vertex $K \in \tilde{Q}$ and let $W = |\mathcal{L} \cup \mathcal{O}|$ denote the total weight of the vertices in $\tilde{Q}$. Since $\tilde{Q}$ has bounded genus, by Theorem 6.1, it has a weighted separator of size $O\left(\sqrt{g\,|\mathcal{K}'|}\right)$. That is, a set $S \subseteq V(\tilde{Q})$ of size $O\left(\sqrt{g\,|\mathcal{K}'|}\right)$ s.t. $V(\tilde{Q}) \setminus S$ separates $\tilde{Q}$ into disconnected components that can be partitioned into two groups A and B with no edges between them, and that $w(A), w(B) \leq \alpha W$, for some constant $0 < \alpha < 1$.

Since $\tilde{Q}$ is an intersection support, each $L \in \mathcal{L}$ and $O \in \mathcal{O}$ induce a connected subgraph of $\tilde{Q}$. Therefore, S yields a separator S' of Ξ consisting of the subgraphs in $\mathcal{L} \cup \mathcal{O}$ that intersect the subgraphs corresponding to vertices in S.

Now, we bound $|S'|$. Consider an arbitrary subgraph $K' \in \mathcal{K}'$. Since both $\mathcal{L}$ and $\mathcal{O}$ are feasible solutions, $1 \leq |\mathcal{L}_{K'}| \leq \text{cap}(K')$ and $1 \leq |\mathcal{O}_{K'}| \leq \text{cap}(K')$. Hence, $|\mathcal{L}_{K'} \cup \mathcal{O}_{K'}| \leq 2\,\text{cap}(K') \leq 2\Delta$. Therefore, any vertex $K' \in \tilde{Q}$ is contained in at most 2Δ subgraphs of $\mathcal{L} \cup \mathcal{O}$. Hence, $|S'| = O\left(\Delta\sqrt{g\,|\mathcal{K}'|}\right)$. Since $|\mathcal{K}'| = O\big(\Delta\big(g + |\mathcal{L} \cup \mathcal{O}|\big)\big)$, this implies that $|S'| = O\left(\Delta^{3/2}\sqrt{g\,(g + |\mathcal{L} \cup \mathcal{O}|)}\right)$.

We finally show that S' is a balanced separator. Since each component of $\tilde{Q} \setminus S$ corresponds to a connected component of $\Xi \setminus S'$, it follows that S' is a balanced separator since S is a balanced separator.

Since Ξ satisfies both the local search property and the sub-linear separator property, it follows that there is a PTAS via the local search framework and the framework in [CH12; MR10]. ∎

For a graph system $(G, \mathcal{H})$, let $\text{cap} : V(G) \to \mathbb{N}$ be a capacity function on $V(G)$. The **Capacitated $\mathcal{H}$-packing** problem is to find a maximum cardinality collection $\mathcal{H}' \subseteq \mathcal{H}$ such that for each $v \in V(G)$, $|\mathcal{H}'_v| \leq \text{cap}(v)$.

Let $\text{cap} : \mathcal{H} \to \mathbb{N}$ be a capacity function on $\mathcal{H}$. Then the **Capacitated Vertex Packing** problem is to find the largest cardinality subset $V' \subseteq V(G)$ s.t. $|H \cap V'| \leq \text{cap}(H)$ for all $H \in \mathcal{H}$.

As a consequence of Theorem 6.2, we get the following corollaries.

Corollary 6.3. Let $(G, \mathcal{H})$ be a cross-free graph system of genus g for some constant g, and let $\Delta > 0$ be an absolute constant. Let $\text{cap} : V(G) \to \mathbb{N}$ be a capacity function such that $\text{cap}(v) \leq \Delta$ for all $v \in V$. Then there is a PTAS for the **Capacitated $\mathcal{H}$-Packing** problem via the local search framework.

Proof. Construct an intersection system $(G, \mathcal{H}, \mathcal{K})$, where $\mathcal{K}$ consists of singleton subgraphs $\{v\}$ for each $v \in V(G)$, with capacity $\text{cap}(v)$. The subgraphs in $\mathcal{K}$ are trivially cross-free. Therefore, by Theorem 6.2, there is a PTAS for the **Capacitated $\mathcal{H}$-Packing** problem for a cross-free system $(G, \mathcal{H})$ of genus g. ∎

Corollary 6.4. Let $(G, \mathcal{H})$ be a cross-free graph system of genus g for some constant g, and let $\Delta > 0$ be an absolute constant. Let $\text{cap} : \mathcal{H} \to \mathbb{N}$ be a capacity function such that $\text{cap}(H) \leq \Delta$ for all $H \in \mathcal{H}$. Then there is a PTAS for the **Capacitated Vertex Packing** problem via the local search framework.

Proof. Construct an intersection system $(G, \mathcal{V}, \mathcal{H})$, where $\mathcal{V}$ consists of singleton subgraphs $\{v\}$ for each $v \in V(G)$, and each $H \in \mathcal{H}$ has capacity $\text{cap}(H)$. Again, the intersection system thus defined is cross-free and has genus g. Therefore, by Theorem 6.2, there is a PTAS for the **Capacitated Vertex Packing** problem for $(G, \mathcal{H})$. ∎

▷ Generalized Covering.

Now we show a PTAS for the **Generalized Covering** problem.

Theorem 6.5. Let $(G, \mathcal{H}, \mathcal{K})$ be a cross-free intersection system of genus g, where g is bounded above by a constant. Then there is a PTAS for the **Generalized Covering** problem via the local search framework.

Proof. Let $\mathcal{L} \subseteq \mathcal{H}$ be a solution returned by the local search framework and let $\mathcal{O} \subseteq \mathcal{H}$ be an optimal solution. We can assume that $\mathcal{L} \cap \mathcal{O} = \emptyset$. Otherwise, we can remove the subgraphs in $\mathcal{L} \cap \mathcal{O}$, and argue instead about the subgraphs in $\mathcal{L}' = \mathcal{L} \setminus (\mathcal{L} \cap \mathcal{O})$ and $\mathcal{O}' = \mathcal{O} \setminus (\mathcal{L} \cap \mathcal{O})$.

Consider the intersection system $(G, \mathcal{L} \cup \mathcal{O}, \mathcal{K})$. By Theorem 2.5, there is an intersection support $\tilde{Q}$ on $\mathcal{L} \cup \mathcal{O}$ of genus at most g. From $\tilde{Q}$, extract the bipartite graph Ξ on $\mathcal{L} \cup \mathcal{O}$. We show that Ξ satisfies the local search property.

Since $\tilde{Q}$ is a support, for each $K \in \mathcal{K}$, there is an $L \in \mathcal{L}_K$ adjacent to an $O \in \mathcal{O}_K$ in $\tilde{Q}$, and hence L and O are adjacent in Ξ. Therefore, for any $\mathcal{L}' \subseteq \mathcal{L}$, $(\mathcal{L} \setminus \mathcal{L}') \cup N(\mathcal{L}')$ is again a feasible solution, where $N(\mathcal{L}')$ refers to the neighborhood of $\mathcal{L}'$ in Ξ. By

Theorem 6.1, $\tilde{Q}$ has a separator of size $O\left(\sqrt{g\,|\mathcal{L} \cup \mathcal{O}|}\right)$, and hence a such a separator for Ξ. It follows that Ξ satisfies the local search property. Therefore, there is a PTAS for the Generalized Covering problem via the local search framework. ■

▷ Independent Set, Dominating Set and Vertex Cover.

For the Independent Set and Dominating Set problems, the algorithm follows directly via local search. For the Vertex Cover problem, we use the well-known half-integrality [NTJ75] of the standard LP-relaxation for Vertex Cover. That is, there is an optimal solution to the LP $\min\{\sum x_v : x_u + x_v \geq 1 \ \forall \{u,v\} \in E, x_u \geq 0\}$, where the x_u take values in $\{0, 1/2, 1\}$. The vertices whose values are either 1 or $1/2$ form a feasible vertex cover. We remove the vertices of weight 0, take all vertices of weight 1, and in the graph induced on the vertices of weight $1/2$, it follows that any vertex cover requires at least $1/2$ of the vertices. Therefore, a $(1-\epsilon)$-approximation to the Independent Set problem on this induced subgraph implies a $(1+\epsilon)$-approximation to the Vertex Cover problem. These ideas have been observed previously by Bar-Yahuda et al. [BYHR11] to obtain a $3/2$-approximation for the Vertex Cover problem on the intersection graph of axis-aligned rectangles.

Corollary 6.6. Let $(G, \mathcal{H})$ be a cross-free graph system of genus g for some constant g, and $\mathcal{D}$ denote the intersection graph. Then, the following problems on $\mathcal{D}$ admit a PTAS: (a) Independent Set, (b) Vertex Cover, and (c) Dominating Set.

Proof. (a) The Independent Set problem is equivalent to the Set Packing problem with $\text{cap}(v) = 1$ for all $v \in V(G)$ and hence a PTAS follows by Corollary 6.3.

(b) For the Vertex Cover problem, we know from the Nemhauser-Trotter theorem [NTJ75] that the standard LP is $1/2$-integral. We only need to find the subgraphs whose LP-weight is $1/2$. Let $\mathcal{H}'$ be the subgraphs of weight $1/2$. For $\mathcal{H}'$, a vertex cover has size at least $|\mathcal{H}'|/2$, since assigning $1/2$ is an optimal solution for the LP-relaxation on the graph induced on the vertices in $\mathcal{H}'$. Since the complement of a vertex cover is an independent set, using (a) we compute a $(1-\epsilon)$-approximation to the Independent Set problem on $\mathcal{H}'$ and take its complement. This immediately implies a $(1+\epsilon)$-approximation to the Vertex Cover problem as the solution VC on $\mathcal{H}'$ that is the complement of the $(1-\epsilon)$-approximation to the independent set computed satisfies $|VC| \leq |\mathcal{H}'| - (1-\epsilon)|I| \leq |\mathcal{H}'| - (1-\epsilon)|\mathcal{H}'|/2$. This implies that $|VC| \leq (1+\epsilon)|\mathcal{H}'|/2 \leq (1+\epsilon)|VC^*|$, where $|VC|^*$ denotes the size of an optimal vertex cover for the graph induced on the vertices of weight $1/2$.

(c) Consider the cross-free intersection system $(G, \mathcal{H}, \mathcal{H})$. There is an intersection support $\tilde{Q}$ of genus at most g. $\tilde{Q}$ satisfies the local search conditions and therefore, the Dominating Set problem on the intersection graph $\mathcal{D}$ admits a PTAS via the local search framework.

■

Summary Table

The table below summarizes the underlying auxiliary hypergraph and the variant of support required to obtain a PTAS for the problems considered in this sub-section.

Problem	**Underlying aux. hyp.**	**Variant of support**
Point Packing	primal	weak bipartite
Hitting Set	primal	weak bipartite
Set Packing	dual	weak bipartite
Set Cover	dual	weak bipartite
Vertex Cover	primal	weak bipartite
Independent Set	dual	weak bipartite
Dominating Set	intersection	weak bipartite
Gen. Covering	intersection	weak bipartite
Cap. Vertex Packing	primal	support
Cap. $\mathcal{H}$-Packing	dual	support
Gen. Cap. Packing	intersection	support

Table 6.1: Support variants required for PTAS in various packing and covering problems.

§ 6.2.2 Regions on an oriented surface

In this section, we give a natural family of geometric hypergraphs that results in cross-free graph systems, and hence the results in the previous section follow for these hypergraphs.

A simple Jordan curve is the image of an injective map from $\mathbb{S}^1$ to $\mathbb{R}^2$. A set of Jordan curves is a set of *pseudocircles* if for any two curves, their boundaries intersect zero or two times. By the Jordan curve theorem (Ch. 2 in [MT01]), a Jordan curve divides the plane into two regions. For a Jordan curve γ, we call the bounded region the *interior* of γ and the unbounded region the *exterior* of γ.

A set of bounded regions $\mathcal{D}$ is a set of *pseudodisks* if their boundaries are pseudocircles. Note that each $D \in \mathcal{D}$ defines a simply connected region [2] such that for any $D, D' \in \mathcal{D}$, $D \setminus D'$ and $D' \setminus D$ are connected.

[2] A region R is simply connected if any simple loop can be continuously contracted to a point so that the loop lies in R throughout the contraction process. Intuitively, a simply connected region does not have *holes*.

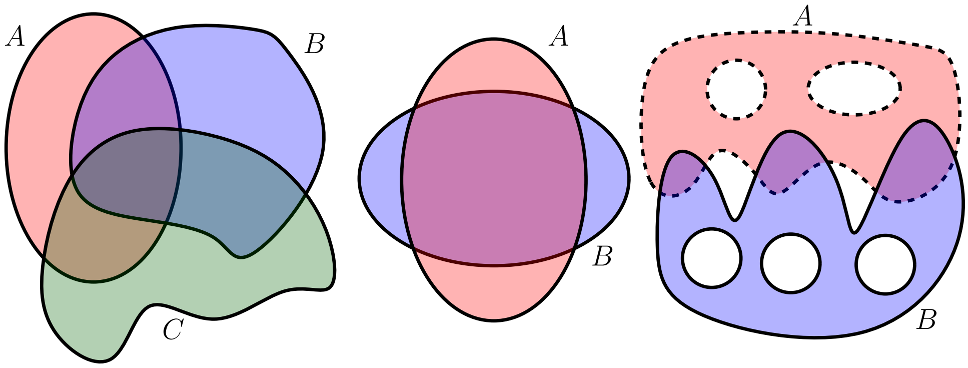


Figure 6.1: An arrangement of Pseudodisks (left), Piercing (middle), and Non-piercing (right) regions in the plane.

If we only insist on the last property, we obtain a set of *non-piercing regions*, a generalization of pseudodisks. A set $\mathcal{D}$ of bounded, connected regions in the plane is a set of non-piercing regions if for any $D, D' \in \mathcal{D}$, $D \setminus D'$ and $D' \setminus D$ are connected (See [RR20] for a formal definition). However, we impose the restriction that the boundaries of D and D' intersect a finite number of times for any $D, D' \in \mathcal{D}$. Note that for non-piercing, the sets in $\mathcal{D}$ need not be simply connected. Fig. 6.1 shows examples of pseudodisks, piercing regions, and non-piercing regions in the plane. Pseudodisks and non-piercing regions generalize several well-studied geometric objects in the plane, such as disks, squares, homothets of a convex set, etc.

Generalizing the previous work [BR+18; CH12; MR10], Raman and Ray [RR20] showed that the intersection hypergraph of non-piercing regions (Definition 6.19) in the plane admits a planar support. However, their results require the non-piercing regions to be in *general position*. That is, the boundaries of any pair of regions intersect only at a finite number of points where they cross, and the boundaries of no three regions cross at a common point. In addition, they require that the points in the input are at some positive distance away from the boundary of the regions. In particular, the non-piercing assumption does not include regions that share a part of their boundary, or have points on the boundary of the regions, such as the regions shown in Fig. 6.2. To handle regions of this type, we introduce the notion of *weakly non-piercing regions* defined as follows.

> **Definition 6.17** (Weakly non-piercing regions)**.** A set $\mathcal{R}$ of connected regions on an oriented surface is said to be weakly non-piercing if for each pair of regions $R, R' \in \mathcal{R}$, either $R \setminus R'$ or $R' \setminus R$ is connected.

In the Conclusion section of [RR20], the authors show an example of a set of weakly non-piercing regions in the plane that does not admit a planar support. We show that if we additionally impose the condition that the regions are *simply connected* and embedded on a surface of genus g, there is a support of genus at most g.

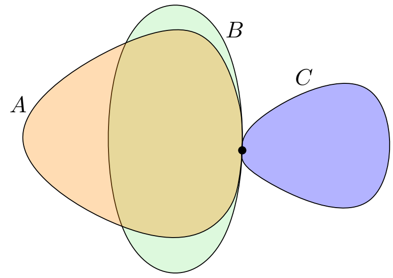


Figure 6.2: The subgraphs A colored orange, B colored green, and C colored blue are touching. Observe that $A \setminus B$ is connected, while $B \setminus A$ induces two connected components. There is an input point on the boundary of the three regions. The regions are weakly non-piercing, but are not non-piercing.

Let $\mathcal{R}$ be a collection of weakly non-piercing regions on a surface of genus g. Consider the boundaries ∂R and $\partial R'$ of two regions R and R' in $\mathcal{R}$, respectively. Then, $\partial R \cap \partial R'$ consists of a finite collection of isolated points and arcs. If at an isolated point, the boundaries of the regions cross, then we call this point a *crossing point*. Otherwise, we call it a *touching point*. Equivalently, a point $p \in \partial R \cap \partial R'$ is a touching point if in an arbitrarily small ball B around p, the boundaries ∂R and $\partial R'$ can be modified by an arbitrarily small amount so that ∂R and $\partial R'$ are disjoint in B. In the following, we define the *dual-arrangement graph* for a collection of connected regions on a surface.

> **Definition 6.18** (Dual-arrangement graph)**.** Given a set $\mathcal{R}$ of connected regions on an oriented surface Σ. The dual-arrangement graph G has a vertex for each cell of the arrangement, and two vertices are adjacent if and only if the corresponding cells share an arc of the boundary of a region in $\mathcal{R}$, or share a touching point.

It is easy to see that the dual-arrangement graph is embeddable on the same surface Σ. Since the regions in $\mathcal{R}$ are connected, it follows that for any region $R \in \mathcal{R}$ the subgraph of G induced by cells in R, is connected. This defines a graph system $(G, \mathcal{R})$, where each region in $\mathcal{R}$ defines a connected subgraph of G induced by the cells contained in it.

We frequently used the notion of an intersection hypergraph, but for completeness, we redefine it in the geometric setting.

> **Definition 6.19** (Intersection hypergraph of regions)**.** Let $\mathcal{H}$ and $\mathcal{K}$ be two families of connected regions on an oriented surface. The intersection hypergraph defined

by $\mathcal{H}$ and $\mathcal{K}$ is the hypergraph that has a vertex for each $H \in \mathcal{H}$, and a hyperedge for each $K \in \mathcal{K}$ defined by $\mathcal{H}_K = \{H \in \mathcal{H} : H \cap K \neq \emptyset\}$.

The following result shows that for an intersection hypergraph defined by weakly non-piercing regions on a surface Σ, there is an equivalent cross-free intersection system embedded on Σ.

Theorem 6.7. Let $\mathcal{H}$ and $\mathcal{K}$ be two families of simply connected weakly non-piercing regions on an oriented surface Σ of genus g. Then, we can define an embedded cross-free intersection system $(G, \mathcal{H}', \mathcal{K}')$ of genus g and a bijection between $\mathcal{H}$ and $\mathcal{H}'$, and between $\mathcal{K}$ and $\mathcal{K}'$ so that the intersection hypergraph defined by $\mathcal{H}$ and $\mathcal{K}$ is isomorphic to that defined by $\mathcal{H}'$ and $\mathcal{K}'$.

Proof. Let $\mathcal{R} = \mathcal{H} \cup \mathcal{K}$. The arrangement of the regions of $\mathcal{H} \cup \mathcal{K}$ partition the surface into *vertices*, *edges* and *cells*. There are two kinds of vertices in the arrangement: when the boundaries of two regions cross, the intersection point is called a *crossing vertex*. For each region $R \in \mathcal{R}$ we subdivide its boundary into arcs such that for each arc a on the boundary of R, and for any two points p, q in the interior of a, the boundaries of set of regions of $\mathcal{R}$ containing p on its boundary, and the set of regions in $\mathcal{R}$ containing q on its boundary are identical. We add the end-points of all such arcs a to the set of vertices, and we call these the *touching vertices*. The arc between two consecutive vertices on the boundary of a region are the edges of the arrangement. For regions in $\mathcal{R}$ not intersected by other regions, the edge contributed by this region is its boundary. Finally, the connected regions obtained by removing the vertices and edges in the arrangement are the cells of the arrangement.

The vertices and edges of the arrangement define a graph G'. To obtain a host graph G, we further subdivide each edge of G' by a vertex. Next, we put a vertex in the interior of each cell in the arrangement. We join the vertex corresponding to a cell to the vertices on its boundary via non-crossing simple arcs that lie in the cell. Since each cell is connected, we can do so. The graph G is clearly embedded on the surface. See Fig. 6.3 for an example.

Consider a region $R \in \mathcal{R}$. Since R is connected, by construction, it defines a connected subgraph of G. Further, since R is simply connected, its boundary is a Jordan curve, and by construction is mapped to a simple cycle C_R in G s.t. this cycle separates the vertices of G that lie in the subgraph corresponding to R from the vertices of G that do not.

We now show that the regions in $\mathcal{H}$ induce cross-free subgraphs of G. The fact that the regions in $\mathcal{K}$ induce cross-free subgraphs of G will then follow analogously. Abusing notation, we use H to also denote the subgraph of G induced on the vertices in H. For contradiction, suppose two subgraphs $H, H' \in G$ corresponding respectively, to regions $H, H' \in \mathcal{H}$ are crossing. Then the reduced graph (See Definition 2.3) $R_G(H, H')$ contains a vertex v and four edges in cyclic order, to vertices v_1, v_2, v_3, v_4 s.t. $v_1, v_3 \in H \setminus H'$ and $v_2, v_4 \in H' \setminus H$. Since the corresponding regions H, H' are weakly non-piercing, either $H \setminus H'$ or $H' \setminus H$ is connected. Suppose, without loss of generality, the former holds. Then in G, there is a path P from v_1 to v_3 that lies entirely in the subgraph $H \setminus H'$. Adding the edges $\{v_1, v\}$ and $\{v, v_3\}$ to P, we obtain a simple cycle C, all of whose vertices lie in H. Since the boundary of the region H defines a simple cycle C_H in G separating the vertices in H from those that do not lie in H, it implies that C lies in the subgraph defined by H. But this implies that either v_2 or v_4 also lies in H, contradicting the assumption that H and H' are crossing. Therefore, the subgraphs defined by regions in $\mathcal{H}$ induce a cross-free collection of subgraphs in G, and analogously, so do the subgraphs defined by the regions in $\mathcal{K}$. Hence, $(G, \mathcal{H}', \mathcal{K}')$ is a cross-free intersection system, where $\mathcal{H}'$ and $\mathcal{K}'$ are respectively, the subgraphs in G corresponding to the regions in $\mathcal{H}$ and $\mathcal{K}$.

Moreover, by construction, any two regions $R_1, R_2 \in \mathcal{H} \cup \mathcal{K}$ intersect if and only if the corresponding subgraphs $R_1', R_2' \in \mathcal{H}' \cup \mathcal{K}'$ share a vertex in G. This implies the intersection hypergraph defined by $\mathcal{H}$ and $\mathcal{K}$, and that defined by $(G, \mathcal{H}', \mathcal{K}')$ are isomorphic. ■

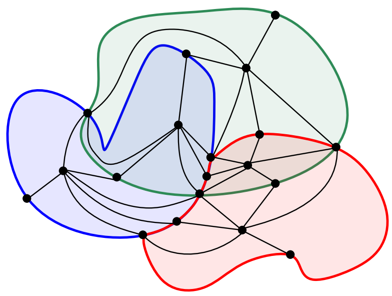


Figure 6.3: Construction of cross-free graph system from simply connected weakly non-piercing regions on a surface.

In combination with the results for packing and covering problems discussed in Section 6.2.1, Theorem 6.7 above implies a PTAS for all such problems defined by simply connected weakly non-piercing regions on an oriented surface of bounded genus.

§ 6.2.3 Packing and covering in non-piercing hypergraphs

Apart from the packing and covering problems defined on cross-free hypergraphs, we highlight similar results for hypergraphs that arise from non-piercing subgraphs or non-piercing geometric objects in the plane. The most natural non-piercing subgraphs are cliques in a graph.

Consider the following problem: Given a graph G of treewidth t and a parameter $k \in \mathbb{N}$, find a smallest subset of vertices that hits all cliques of size at least k. A generalized version of this problem is – Given a set $\mathcal{H}$ of cliques and a set $\mathcal{K}$ of cliques, find a smallest subset of cliques in $\mathcal{H}$ that hit all the cliques in $\mathcal{K}$. Similarly, for the Generalized Packing, find a maximum subset $\mathcal{H}' \subseteq \mathcal{H}$ such that no clique in $\mathcal{K}$ intersects more than one clique in $\mathcal{H}'$. As a second example, let $\mathcal{H}$ be a collection of *non-crossing* paths in G, where we say that two paths P_1 and P_2 are non-crossing if once the paths meet at a vertex, they do not get separated. Non-crossing paths are, by definition, non-piercing. The goal is to select the smallest number of edge-disjoint, or vertex-disjoint paths from $\mathcal{H}$. To be precise, we state below the results for the generalized versions of these problems for cliques only, which are a direct implication of Theorem 4.4, and the local search framework. The proofs rely on the same reasoning as Theorems 6.2 and 6.5, respectively, so we omit them.

Theorem 6.8. Let $(G, \mathcal{H}, \mathcal{K})$ be an intersection system of treewidth t, where $\mathcal{H}$ and $\mathcal{K}$ are sets of cliques. Then, the **Generalized Packing** problem admits a PTAS.

Theorem 6.9. Let $(G, \mathcal{H}, \mathcal{K})$ be an intersection system of treewidth t, where $\mathcal{H}$ and $\mathcal{K}$ are sets of cliques. Then, the **Generalized Covering** problem admits a PTAS.

▷ Approximating Hitting Set for axis-aligned rectangles.

We now show that for the hypergraph defined by points and axis-parallel rectangles in the plane, there is an approximation algorithm with a small approximation factor. In this context, the Hitting Set problem has input a set P of points and a set $\mathcal{R}$ of axis-parallel rectangles. The objective is to select a smallest subset $P' \subseteq P$ such that $P' \cap R \neq \emptyset$ for each $R \in \mathcal{R}$. We use the fact that there is a planar support for non-piercing axis-parallel rectangles (Theorem 5.2) and the fact that the piercing graph on the rectangles is a comparability graph (Theorem 5.3).

Note that for this result, we only require the existence of a planar support for points with respect to non-piercing rectangles, and we do not need to explicitly construct them. Such a result was already known, and was proved by [PR08], which was used by Mustafa and Ray [MR10] to show that a simple local search algorithm yields a PTAS for a more general family, namely, the pseudodisks. Our contribution then is only to use the observation in Theorem 5.3, and if the piercing graph has small chromatic number, this implies an approximation algorithm with a small factor for the Hitting Set problem.

Theorem 6.10. Let P be a set of points and $\mathcal{R}$ be a set of axis-parallel rectangles in the plane such that the chromatic number $\chi(H)$ of the piercing graph H is at most k. Then, for any $\epsilon > 0$, there is a $(k+\epsilon)$-approximation algorithm for the Hitting Set problem on $(\mathcal{R}, P)$.

Proof. The piercing graph H of $\mathcal{R}$ can be computed in polynomial time. By Theorem 5.3, H is a comparability graph. Since $\chi(H) \leq k$ and a proper coloring of H with at most k colors can be computed in polynomial time [Gol80b], we can partition $\mathcal{R}$ into at most k families $\mathcal{R}_1, \ldots, \mathcal{R}_k$ s.t. each $\mathcal{R}_i$ consists of non-piercing rectangles.

To compute the hitting set for rectangles $\mathcal{R}$ and points P, we apply a PTAS for the Hitting Set problem [MR10] with rectangles $\mathcal{R}_i$ and point set P. We return the union of the solutions for the k Hitting Set problems, whose size we denote by S. Let OPT denote the optimal size of a hitting set for the input rectangles with points P. Let OPT_i and S_i denote respectively, the size of an optimal hitting set for the rectangles in $\mathcal{R}_i$, and the size of a hitting set returned by the PTAS for $\mathcal{R}_i$. Consider $\epsilon' = \epsilon/k$, where each PTAS returns an ϵ-approximate solution. Then,

$$\begin{aligned} S &\leq S_1 + S_2 + \ldots + S_k \\ &\leq (1+\epsilon')\,\text{OPT}_1 + (1+\epsilon')\,\text{OPT}_2 + \ldots + (1+\epsilon')\,\text{OPT}_k \\ &\leq (k+\epsilon)\,\text{OPT}. \end{aligned}$$

■

§ 6.3 Hardness Results for Crossing Systems

In this section, we show some APX-hardness (see Definition 6.20) results for packing and covering problems for geometric hypergraphs. In the planar case, Chan and Grant [CG14b] proved that for hypergraphs defined by points, and horizontal and vertical slabs, the Set Cover, Hitting Set, Point Packing, and Set Packing problems are APX-hard. However, if the geometric regions are non-piercing, the existence of a planar support in conjunction with the local search framework implies PTAS to all the problems above [RR20].

Unfortunately, the non-piercing property can not be extended beyond planarity. A simple modification of the results of Chan and Grant [CG14b] implies the APX-hardness results for non-piercing graph systems that are not cross-free, and we show this in Theorem 6.13. We believe that the topological condition of being cross-free is essential to obtain PTAS for these problems when the parameter 'genus' of the host graph is bounded above by a constant. Before we prove Theorem 6.13, we outline the

APX-hardness results of Chan and Grant for a family of geometric hypergraphs in the plane.

Definition 6.20 (APX-hard). *APX* is the class of all optimization problems that admit a constant-factor approximation algorithm. An optimization problem Π is *APX-hard* if every problem in the class APX can be approximation-preserving reduced to Π.

By the assumption P $\neq$ NP, an APX-hard problem does not possess a PTAS.

Definition 6.21 (SPECIAL-3SC). Let $V = A \cup W \cup X \cup Y \cup Z$ for disjoint sets $A = \{a_1, \ldots, a_n\}$, $W = \{w_1, \ldots, w_m\}$, $X = \{x_1, \ldots, x_m\}$, $Y = \{y_1, \ldots, y_m\}$ and $Z = \{z_1, \ldots, z_m\}$, where $2n = 3m$. Let $\mathcal{S}$ be the collection of $5m$ subsets of V such that:

(i) For each $t \in [m]$, $\exists\, 1 \leq i < j < k \leq n$ such that the sets $\{a_i, w_t\}$, $\{w_t, x_t\}$, $\{a_j, x_t, y_t\}$, $\{y_t, z_t\}$ and $\{z_t, a_k\}$ lie in $\mathcal{S}$, and

(ii) a_i lies in precisely two sets in $\mathcal{S}$, for all $1 \leq i \leq n$.

Then, SPECIAL-3SC is the Set Cover problem for the hypergraph $\mathcal{H} = (V, \mathcal{S})$.

The authors showed that SPECIAL-3SC is APX-hard via a reduction from a known APX-hard problem *3VC* [AK00], where 3VC is the minimum vertex cover problem for 3-regular graphs.

Lemma 6.2 ([CG14b]). SPECIAL-3SC is APX-hard.

Proof. Let $G = (V, E)$ be an instance of 3VC with $V = \{v_1, \ldots, v_m\}$ and $E = \{e_1, \ldots, e_n\}$. Since G is 3-regular, $3m = 2n$. We will define a hypergraph $\mathcal{H} = (V', \mathcal{S})$ that corresponds to an instance of SPECIAL-3SC such that an optimal solution for *Special-3SC* leads to an optimal solution for 3VC.

The set $V' = A \cup W \cup X \cup Y \cup Z$ is the disjoint union of sets $A = \{a_1, \ldots, a_n\}$, $W = \{w_1, \ldots, w_m\}$, $X = \{x_1, \ldots, x_m\}$, $Y = \{y_1, \ldots, y_m\}$ and $Z = \{z_1, \ldots, z_m\}$. For each vertex $v_t \in V$ and edges e_i, e_j, e_k incident to v_t s.t. $i < j < k$, $\mathcal{S}$ contains the sets $\{a_i, w_t\}$, $\{w_t, x_t\}$, $\{a_j, x_t, y_t\}$, $\{y_t, z_t\}$ and $\{z_t, a_k\}$. This completes the construction of $\mathcal{H}$.

Let σ be a feasible solution to SPECIAL-3SC for $\mathcal{H}$. We construct a feasible solution σ' to the problem 3VC for G. For each $t \in [m]$, add the vertex $v_t \in V$ to σ' iff σ contains at least one of the sets $\{a_i, w_t\}$, $\{a_j, x_t, y_t\}$ or $\{z_t, a_k\}$. Then a vertex $v \in \sigma'$ covers an edge e_ℓ iff σ contains a set cering the vertex a_ℓ.

A set in $\mathcal{S}$ containing an element of A is called a $\mathcal{P}$-set; otherwise, it is called a $\mathcal{Q}$-set. The feasible solution σ is called *segregated* if for each $t \in [m]$, either it contains all the corresponding $\mathcal{P}$-sets and no $\mathcal{Q}$-set, or vice versa. Further, for any optimal solution to SPECIAL-3SC, there is a segregated optimal solution - for $t \in [m]$, if the optimal solution contains a $\mathcal{Q}$ set, we add all $\mathcal{Q}$ sets corresponding to t, otherwise we include all the corresponding $\mathcal{P}$-sets. Then the cost of the updated solution is no more than the optimal solution we started with, and hence is a segregated optimal solution.

It can be seen that the set of segregated optimal solutions to SPECIAL-3SC is in bijection with the set of feasible solutions to 3VC. Further, if OPT and OPT' are the optimal values for SPECIAL-3SC and 3VC respectively, then $OPT = OPT' + 2m$, the lemma follows. ■

Lemma 6.2 implies the following.

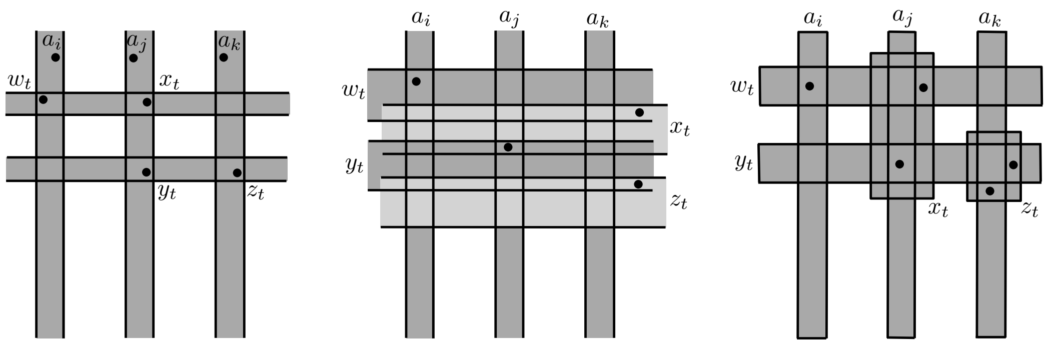
Theorem 6.11 ([CG14b]). The Set Cover and the Hitting Set problems are APX-hard for the hypergraphs defined by the following geometric objects.

1. Points and axis-parallel slabs in $\mathbb{R}^2$.
2. Points and axis-parallel rectangles in $\mathbb{R}^2$ even if the boundaries of each pair of rectangles intersect zero or four times.

Proof. Consider an instant of SPECIAL-3SC for the hypergraph $\mathcal{H} = (V, \mathcal{S})$ as defined in Definition 6.21. We construct an instance of the Set Cover problem for the hypergraphs defined in Items 1 and 2, as shown in Fig. 6.4a.

For each element of A, there is a vertical slab containing it so that all the elements of A appear in a horizontal line. For each $\mathcal{P}$-set in $\mathcal{S}$, i.e., for the sets of the form $\{w_t, x_t\}$ and $\{x_t, y_t\}$, we add a horizontal slab and the elements of the corresponding $\mathcal{P}$-set. By definition, each element of A is contained in exactly two sets in $\mathcal{S}$. Let S_1 and S_2 be respectively the first and second sets in $\mathcal{S}$ that contain $a \in A$. We place the elements of S_1 slightly towards the left to the position of a, and those of S_2 slightly towards the right to the position of a. This completes the construction of the Set Cover problem for hypergraphs in Items 1 and 2. Clearly, any feasible or optimal solution for the Set Cover problem for $\mathcal{H}$ is also a feasible or optimal solution for the given instance of SPECIAL-3SC and conversely. This proves the APX-hardness of the Set Cover problem.

For the Hitting Set problem, we modify the construction to a dual hypergraph. For the hypergraph in Item 1, each element of A corresponds to a vertical slab, and each element of $V \setminus A$ corresponds to a horizontal slab. See Fig. 6.4b. Further, for each $S \in \mathcal{S}$, we place a point that lies in the intersection of the slabs corresponding to the elements of S. By construction, any feasible or optimal solution to the Hitting Set problem for the constructed hypergraph is also a feasible or optimal solution to the given instance of SPECIAL-3SC. The argument for the Hitting Set problem for the hypergraph in Item 2 is similar, and is shown in Fig. 6.4c. ■



(a) Set cover for axis-parallel slabs/rectangles. **(b)** Hitting set for axis-parallel slabs. **(c)** Hitting set for axis-parallel rectangles.

Figure 6.4: APX-hardness of Set Cover/ Hitting Set problems for axis-aligned slabs/rectangles - reduction from the SPECIAL-3SC problem. (The above figures are from [CG14b]).

Now we present the APX-hardness result of Chan and Grant [CG14b] for the packing problem for the hypergraph arising from SPECIAL-3SC.

In the proof of Lemma 6.2 we saw that for every 3-regular graph G with m vertices and n edges, there is a hypergraph associated with the SPECIAL-3SC problem. The converse is also true, i.e., for every instance $\mathcal{H} = (V, \mathcal{S})$ of SPECIAL-3SC, there is an associated 3-regular graph on m vertices since $\mathcal{H}$ satisfies $3m = 2n$ (see Definition 6.21). By Lemma 6.2, we get the following.

Lemma 6.3 ([CG14b])**.** Let $\mathcal{H}$ be a hypergraph associated with the SPECIAL-3SC problem. Then the Set Packing problem on $\mathcal{H}$ is APX-hard.

Proof. Let G be a 3-regular graph and G' be the graph obtained from G by replacing each vertex v by a path on five vertices, and making the first, third, and fifth vertex of the path adjacent to the neighbors of v (in any order). This gives a one-to-one correspondence between the vertices of G' and the sets in $\mathcal{S}$, where $\mathcal{S}$ denotes the set of hyperedges of $\mathcal{H}$. Also, two vertices in G' are adjacent if and only if the corresponding sets in $\mathcal{S}$ share a vertex of $\mathcal{H}$. Note that each vertex of G' has degree at most 3.

It is easy to see that the Set Cover problem on $\mathcal{H}$ is equivalent to the vertex cover problem on G'. The complement of a vertex cover is an independent set of vertices. By the earlier argument, any independent set of vertices in G' gives any feasible solution to the Packing Problem for $\mathcal{H}$. Since vertex cover is APX-hard for graphs of degree at most 3 [AK00], the lemma follows. ■

The following result is a direct implication of Lemma 6.3 and Theorem 6.11.

Theorem 6.12 ([CG14b]). The Point Packing and Set Packing problems are APX-hard for the hypergraphs defined by the following geometric objects.

1. Points and axis-parallel slabs in $\mathbb{R}^2$.
2. Points and axis-parallel rectangles in $\mathbb{R}^2$ even if the boundaries of each pair of rectangles intersect zero or four times.

Now we are ready to prove the main result of this section.

Theorem 6.13. There exist non-piercing graph systems $(G, \mathcal{H})$ which are not cross-free, with G embedded on the torus, such that the Set Cover problem is APX-hard. Similarly, the Hitting Set, Point Packing, and Set Packing problems on such a set system are APX-hard.

Proof. The proof follows directly from the corresponding APX-hardness proof by Chan and Grant [CG14b] presented for Theorems 6.11 and 6.12. We only sketch the modification required. We highlight the construction for the Set Cover problem, and the proof for the Hitting Set and Point/Set Packing problems will be followed similarly.

Consider the Set Cover problem: Given a set P of points, and a set $\mathcal{S} = \mathcal{S}_h \cup \mathcal{S}_v$ of axis-aligned slabs in $\mathbb{R}^2$, where $\mathcal{S}_h$ is a set of horizontal slabs and $\mathcal{S}_v$ is a set of vertical slabs. By Theorem 6.11, it is APX-hard to select a minimum cardinality subset of $\mathcal{S}$ to cover P. To obtain the claimed APX-hardness proof on the torus for non-piercing regions, we embed this construction on a torus, and then modify the boundary of each region in $\mathcal{S}_h$ to be a pair of parallel non-separating closed curves parallel to the hole. Similarly, we map each vertical slab in $\mathcal{S}_v$ to a region bounded by two parallel non-separating closed curves perpendicular to the hole. Let $\mathcal{S}'$ be the set of modified regions on the torus. Note that the regions in $\mathcal{S}$ are not non-piercing, but the modified regions in $\mathcal{S}'$ are non-piercing.

We assume without loss of generality that in the arrangement of regions in $\mathcal{S}'$, a cell containing a point of P contains exactly one point. Let G be the dual-arrangement graph of the regions in $\mathcal{S}'$. We assign blue colour to the vertices of G that correspond to non-empty cells, and assign red colour to the remaining vertices. Let $\mathcal{H}$ denote the set of subgraphs of G defined by the regions $\mathcal{S}'$. In $(G, \mathcal{H})$, the subgraphs are non-piercing, but are not cross-free (at the vertices that lie in the intersection of a vertical and a horizontal slab). This completes the construction of the required graph system. The APX-hardness of the Set Cover problem now follows similarly to the proof of Theorem 6.11. ∎

§ 6.4 Hypergraph Coloring

Hypergraph coloring generalizes the classical notion of graph coloring, and is defined below.

Definition 6.22 (k-coloring)**.** A *proper k-coloring* of a hypergraph $\mathcal{H} = (X, \mathcal{E})$ is an assignment $\chi : X \to \{1, 2, \ldots, k\}$ such that for each $E \in \mathcal{E}$ containing at least two vertices, E is not monochromatic, i.e., there are $u, v \in E$ satisfying $\chi(u) \neq \chi(v)$.

A hypergraph $\mathcal{H}$ is said to be *k-colorable*, if there is a proper k-coloring of $\mathcal{H}$. The *chromatic number* of $\mathcal{H}$ is the minimum integer k such that $\mathcal{H}$ is proper k-colorable. In the following, we use the term coloring to mean proper coloring, and we assume without loss of generality that each hyperedge contains at least two elements.

Keller and Smorodinsky [KS18] showed that the intersection hypergraph of disks in the plane is 4-colorable, i.e., a coloring with 4 colors such that no hyperedge is monochromatic. This was generalized by Keszegh [Kes20] for pseudodisks, which was further generalized by Raman and Ray [RR20] to show that the intersection hypergraph of non-piercing regions is 4-colorable. We generalize these results to regions defined on higher genus surfaces. Again, since the intersection hypergraphs generalize primal and dual hypergraphs, we state our result only for the general setting. We first show the argument for a cross-free graph system.

Theorem 6.14. Let $(G, \mathcal{H}, \mathcal{K})$ be a cross-free graph system of genus g. The intersection hypergraph is $\left(\frac{7+\sqrt{1+48g}}{2}\right)$- colorable.

Proof. By Theorem 2.5, $(G, \mathcal{H}, \mathcal{K})$ has an intersection support $\tilde{Q}$ of genus at most g. Now, $\chi(\tilde{Q}) \leq \left(\frac{7+\sqrt{1+48g}}{2}\right)$ [RY68] (see also Chapter 8 of [MT01]). Since $\tilde{Q}$ is a support, for each $K \in \mathcal{K}'$, there is an edge between some two subgraphs $H, H' \in \mathcal{H}_K$. Therefore, any proper coloring of $\tilde{Q}$ is also a proper coloring of the underlying intersection hypergraph. Thus, the chromatic number of the intersection hypergraph is upper bounded by $\chi(\tilde{Q})$, the result follows. ∎

Note that the above proof requires $\tilde{Q}$ to be a weak support, and not necessarily a support. Now, Theorems 6.7 and 6.14 together imply the following.

Corollary 6.15. Let $\mathcal{H}$ and $\mathcal{K}$ be two families of simply connected weakly non-piercing regions on an oriented surface of genus g. Then the intersection hypergraph defined by $\mathcal{H}$ and $\mathcal{K}$ is $\left(\frac{7+\sqrt{1+48g}}{2}\right)$- colorable.

Ackerman et al. [AKP20] considered a notion of $ABAB$-free hypergraphs, which is defined as follows: a hypergraph $(X, \mathcal{S})$ is $ABAB$-free if there exists a linear ordering $x_1 < \ldots < x_n$ of X such that for any pair of hyperedges $A, B \in \mathcal{S}$, there are no four elements $x_i < x_j < x_k < x_\ell$ such that $x_i, x_k \in A \setminus B$ and $x_j, x_\ell \in B \setminus A$. The notion of $ABAB$-free hypergraphs is equivalent to the notion of $abab$-free hypergraphs where the vertices are in a cyclic order. Indeed, if there exists a linear ordering $x_1 < \ldots < x_n$ that is $ABAB$-free, then the cyclic order $x_1 < \ldots < x_n < x_1$ is $abab$-free, and similarly, if $x_1 < \ldots, x_n < x_1$ is a cyclic order that is $abab$-free, then $x_1 < \ldots < x_n$ is an $ABAB$-free linear order. We will revisit this notion and its generalization in the next chapter.

An arrangement of pseudodisks in the plane is said to be a *stabbed* if all of them contain a common point. The authors in [AKP20] show that $ABAB$-free hypergraphs are equivalent to hypergraphs with a *stabbed pseudodisk representation*. In other words, each $S \in \mathcal{S}$ is mapped to a simply connected bounded region D_S containing the origin such that boundaries of any two regions intersect at most twice, and each $x \in X$ is mapped to a point $p_x \in \mathbb{R}^2$ such that $x \in S$ iff $p_x \in D_S$.

Let $\mathcal{D} = \{D_S : S \in \mathcal{S}\}$ be a stabbed pseudodisk arrangement, and P be a set of points in the plane. The authors show that we can add additional pseudodisks $\mathcal{D}'$ such that (i) each $D' \in \mathcal{D}'$ contains exactly 2 points of P, (ii) $\mathcal{D} \cup \mathcal{D}'$ is a pseudodisk arrangement, and (iii) Each $D \in \mathcal{D}$ such that $|D \cap P| \geq 3$ contains a pseudodisk $D' \in \mathcal{D}'$. The graph on P whose edges are defined by $\mathcal{D}'$ is called the *Delaunay graph* of the arrangement.

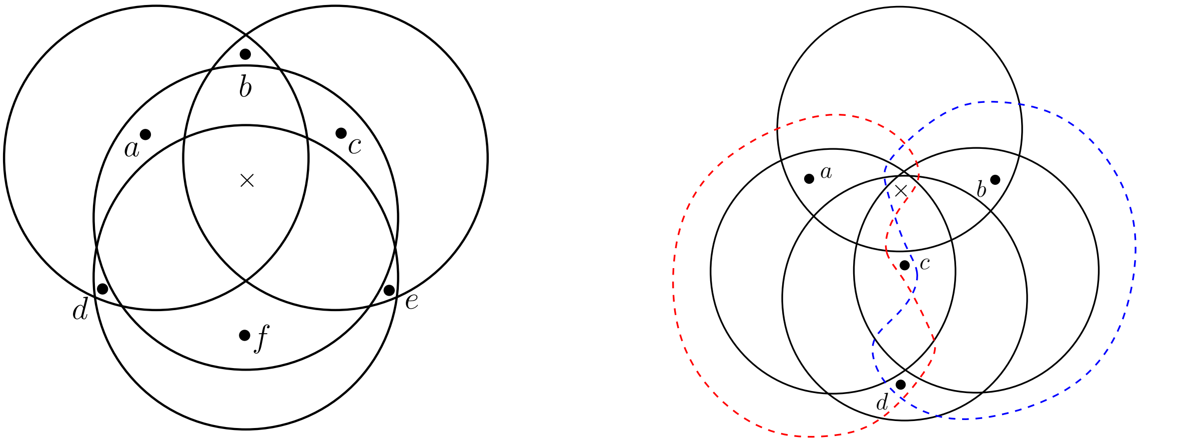
Our result, namely Lemma 3.5 in Chapter 3 is stronger. A Delaunay graph ensures that for each pseudodisk $D \in \mathcal{D}$, the induced subgraph on the elements in D is non-empty, i.e., the Delaunay graph is a weak support for the geometric hypergraph $(P, \mathcal{D})$. However, Lemma 3.5 implies that the hypergraph admits an outerplanar support. This also implies their result that an $ABAB$-free hypergraph is 3-colorable.

At the outset, it seems like the results of Ackerman et al. [AKP20] (especially their proof of Lemma 2.1) can be used to prove Lemma 3.5. However, there is a subtle difference between the two. The authors show that there is a 2-element hyperedge, or equivalently a non-blocking diagonal that can be added between two elements of a hyperedge, but our result shows that such a diagonal can be added between two disjoint runs of a hyperedge (subgraph) which is a more stringent condition.

It was asked in the conclusion section of [AKP20] whether the dual of an $ABAB$-free hypergraph is 3-colorable. In Fig. 6.5a, we answer this question in negative – even if the regions are defined by unit disks in the plane. The figure shows a stabbed pseudodisk arrangement, and hence corresponds to an $ABAB$-free hypergraph with points as the vertices and disks as hyperedges. It can be easily verified that the dual support is K_4, which is not 3-colorable.

Consider the following natural extension of the result of Ackerman et al. [AKP20]: Call an arrangement of non-piercing regions *stabbed* if their intersection is non-empty.

Given a collection of stabbed non-piercing regions in the plane, does there exist a coloring of the points with 3 colors such that no region is monochromatic? We answer this question again in the negative by giving a counter-example (see Fig. 6.5b). It is easy to check that in this case again, the primal support graph is K_4, and therefore the hypergraph is not 3-colorable.



(a) Dual hypergraph: very point $a, b, \ldots, f$ is contained in two disks.

(b) Primal hypergraph: Every region contains two points.

Figure 6.5: Stabbed hypergraphs of disks (dual) and non-piercing regions (primal) requiring four colors.

§ 6.5 Concluding Remarks and Open Questions

In this chapter, we studied applications of support and its variants to several packing, covering, and coloring problems for hypergraphs defined by subgraphs of a host graph, as well as for hypergraphs that arise from geometry. This generalizes the previous work from the planar setting to higher genus surfaces. In particular, we established PTAS for the Generalized Capacitated Packing problem and the Generalized Covering problem for cross-free graph systems defined on a bounded genus surface by leveraging the fact that they admit sparse supports. We introduced weakly non-piercing regions on an oriented surface, a notion weaker than the notion of non-piercing regions. Hypergraphs defined by these objects can be transformed into cross-free graph systems, thereby admitting PTAS for the packing and covering problems above. We also proved that dropping the cross-free condition leads to APX-hardness of these problems.

For the hypergraph coloring problem, we showed that the intersection hypergraph arising from cross-free graph system of genus g has chromatic number at most $\left(\frac{7+\sqrt{1+48g}}{2}\right)$ since the chromatic number of a hypergraph is upper bounded by the chromatic number of any of its supports.

▷ Limitations.

The results of this chapter come with some limitations, which naturally point to directions for further refinement.

Most of the naturally studied hypergraphs arise from geometry. We saw a strong relation between sparse support and better approximation algorithms. In $\mathbb{R}^2$, the existence of a planar support was known for the intersection hypergraph of non-piercing regions [RR20], and we extended this to surfaces of higher genus (Theorems 2.5 and 6.7) by relaxing the non-piercing condition to weakly non-piercing but with a restriction of being simply connected. The two properties – a) (weakly) non-piercing and b) simply connectedness – thus emerge as crucial for obtaining sparse supports and, consequently, better algorithmic results. Relaxing either of these properties results in a dense support and APX hardness for packing and covering as observed in Example 2.1 and Theorems 6.11 to 6.13. On the other hand, when both properties are satisfied, Theorem 6.7 guarantees the existence of sparse supports and thereby enables PTAS. Fig. 6.6 summarizes the nature of support and algorithmic complexity for geometric hypergraphs defined by various connected regions in $\mathbb{R}^2$ and on higher genus surfaces.

We could not establish a complete characterization of hypergraphs admitting sparse support and better algorithmic results for the optimization problems discussed above. In light of this discussion, we state the following open problem.

> **Open Problem 12.** For hypergraphs satisfying only one of the conditions – (weakly) non-piercing and simply connectedness, what further restrictions should be imposed so as to obtain a sparse support and hence PTAS for the packing and covering problems for the resulting hypergraphs?

We conclude the chapter with a small argument on hypergraph coloring. In the proof of Theorem 6.14 we saw that a hypergraph $\mathcal{H}$ satisfies the following inequality for any weak support Q.

$$\chi(\mathcal{H}) \leq \chi(Q) \tag{6.1}$$

However, this bound can be arbitrarily far from being tight. Indeed, consider the hypergraphs in Examples 2.2 and 2.3. It can be easily verified that in each case, the chromatic number is 2, however, the weak supports are complete graphs. There is a

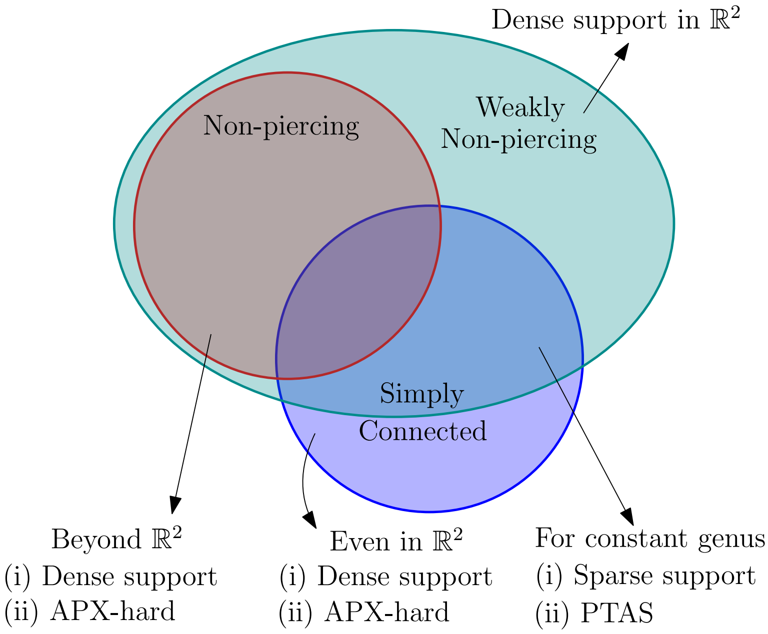


Figure 6.6: Venn diagram of various geometric hypergraphs in orientable surfaces. The figure describes the nature of support and algorithmic complexity for packing and covering problems.

simple catch - remove the unique vertex contained in every hyperedge, the Eq. (6.1) turns into equality for the resulting hypergraph. However, we don't know this in general, and in fact, these examples also show that unlike the classical graph coloring, the hypergraph coloring is not monotone in the sense that the chromatic number of a sub-hypergraph can be arbitrarily larger than that of the parent hypergraph. This motivates the following open problem.

> **Open Problem 13.** Characterize the hereditary family $\mathcal{F}$ of hypergraphs such that for each hypergraph $\mathcal{H} \in \mathcal{F}$, $\chi(\mathcal{H}) = \Theta(\chi(Q))$ for some appropriate weak support Q of $\mathcal{H}$.

Hypergraphs defined by planar non-piercing regions, or those defined by weakly non-piercing simply connected regions on surfaces with constant genus, satisfy $\chi(\mathcal{H}) = \Theta(\chi(Q))$ and thus, $\mathcal{F} \neq \emptyset$. Extending this line of investigation to broader classes of geometric hypergraphs represents a natural direction for further research.

(C) Complexity of Hypergraph Recognitions

Chapter 7

ABAB-Testing Complexity

Abstract

The study of geometric hypergraphs gave rise to the notion of $ABAB$-free hypergraphs. A hypergraph $\mathcal{H}$ is called $ABAB$-free if there is an ordering of its vertices such that there are no hyperedges A, B and vertices v_1, v_2, v_3, v_4 in this order satisfying $v_1, v_3 \in A \setminus B$ and $v_2, v_4 \in B \setminus A$. In this chapter, we prove that it is NP-complete to decide if a hypergraph is $ABAB$-free. We show a number of analogous results for hypergraphs with similar forbidden patterns, such as $ABABA$-free hypergraphs. As an application, we show that deciding whether a hypergraph is realizable as the incidence hypergraph of points and pseudodisks is also NP-complete. The material presented in this chapter is derived from the following publication.
"**The complexity of recognizing $ABAB$-free hypergraphs**".

Table of Contents

§ 7.1 Introduction

Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph with an ordered vertex set and $k \in \mathbb{N}$. We say that edges H and L *form an* $(AB)^k$ *pattern in* $\mathcal{H}$ if there exist $h_1, l_1, h_2, l_2, \ldots, h_k, l_k \in V$ in this order such that $h_i \in H \setminus L$ and $l_j \in L \setminus H$ for all $1 \leq i \leq k$ and $1 \leq j \leq k$.[1] An ordering π of V is said to be $(AB)^k$*-free*, if no pair of edges form an $(AB)^k$ pattern. A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is called $(AB)^k$*-free* if there is an ordering π of V with which $\mathcal{H}$ is $(AB)^k$-free.

Similarly, H and L *form an* $(AB)^{k-1}A$ *pattern in* $\mathcal{H}$ if there exist $2k-1$ vertices $h_1, l_1, h_2, l_2, \ldots h_{k-1}, l_{k-1}, h_k \in V$ in this order such that $h_i \in H \setminus L$ and $l_j \in L \setminus H$ for all $1 \leq i \leq k$ and $1 \leq j \leq k-1$, and we define $(AB)^{k-1}A$*-free* orderings and $(AB)^{k-1}A$*-free* hypergraphs accordingly.

ABA-free hypergraphs were introduced by Keszegh and Pálvölgyi [KP19] to study pseudohalfplane arrangements from an abstract viewpoint, and this enabled them to generalize a result of Smorodinsky and Yuditsky about polychromatic colorings for halfplanes [SY12]. The notion was also generalized to a higher number of alternations in the same paper; for more background, see [AKP20], or for a recent application, see [LN23]. Similar alternations of hypergraphs were also studied independently in an entirely different context to bound the chromatic number of certain generalized Kneser graphs [AH15].

In this chapter, we study the problem of deciding whether an input hypergraph is $(AB)^k$-free or not. This is equivalent to whether it can be realized as a specific geometric hypergraph. More precisely, a hypergraph $\mathcal{H} = (V, \mathcal{E})$ is $(AB)^k$-free exactly if there are maps $\alpha \colon V \to \mathbb{R}^2$ and $\gamma \colon \mathcal{E} \to \mathbf{C}(\mathbb{R})$ (the set of continuous $\mathbb{R} \to \mathbb{R}$ functions) such that $\alpha(v)$ is over the graph of the function $\gamma(E)$ if and only if $v \in E$, and for any two distinct $E, E' \in \mathcal{E}$, the graphs of $\gamma(E)$ and $\gamma(E')$ intersect at most $2k-2$ times. A similar statement is true for $(AB)^kA$-free hypergraphs and $2k-1$ intersections. If $k = 2$, i.e., for $ABAB$-free hypergraphs, there is an equivalent characterization with the interiors of Jordan curves that intersect at most twice and enclose a common point, namely, stabbed pseudodisks; see [AKP20].

A *c-uniform hypergraph* is one with each hyperedge containing exactly c vertices. We note that $ABAB$-freeness of 2-uniform hypergraphs can be decided in polynomial time. In [AKP20], it was shown that $ABAB$-free graphs are outerplanar. The converse is also true: any outerplanar graph is $ABAB$-free. Indeed, we can simply take the vertices in the order they appear along the outer face. (Cut vertices appear more than once along the boundary, we keep an arbitrary one of these.) Any $ABAB$ pattern would correspond to crossing chords, hence the ordering is $ABAB$-free. In fact, Theorem 3.8 also implies that a 2-uniform hypergraph is $ABAB$-free iff it is an outerplanar graph.

[1] Equivalently, H and L form an $(AB)^k$ pattern if writing down respectively the symbol h or l for each vertex contained in only one of the hyperedges, we do *not* get a Davenport–Schinzel sequence of order $2k-2$.

We show the following results.

Theorem 7.1. For any positive integer $k \geq 2$, it is NP-complete to decide if a given hypergraph is $(AB)^k$-free.

Theorem 7.2. For any positive integer $k \geq 2$, it is NP-complete to decide if a given hypergraph is $(AB)^k A$-free.

Let π be any ordering of the vertices of a hypergraph $\mathcal{H}$, and k be any fixed positive integer. It is easy to see that one can decide in polynomial time if $\mathcal{H}$ is $(AB)^k$-free or if it is $(AB)^k A$-free w.r.t. the ordering π. This shows that the decision problems in Theorem 7.1 and Theorem 7.2 are in class NP. Therefore, to prove the theorems, it is enough to show the NP-hardness of these problems.

▷ Organization of the chapter.

In Section 7.2 and Section 7.3 we show that deciding $ABAB$-freeness and deciding $ABABA$-freeness are NP-hard. Then in Section 7.4, we show how Theorem 7.1 and Theorem 7.2 follow from these results. As a geometric application, we show in Section 7.5 that it is NP-complete to decide whether a given hypergraph is realizable as the incidence hypergraph of points and a family of pseudodisks. These results leave the complexity of ABA-freeness undecided, and this is discussed in Section 7.6.

§ 7.2 $ABAB$-freeness is NP-complete

In this section, we show the following special case of Theorem 7.1.

Theorem 7.3. It is NP-hard to decide if a given hypergraph is $ABAB$-free.

For the proof of Theorem 7.3 we need a couple of notations and lemmas. Throughout the proof, we will have a fixed N, and the vertex set of our hypergraph will be $\{1, 2, \dots, N\}$. So, for the following definitions and lemmas, assume that N is fixed. For $S \subseteq \{1, 2, \dots, N\}$ we say that S is an *interval*, if the elements of S are consecutive, and S is a *t-interval* if it is the union of t intervals (some of which might be empty). For $x < y$, let $[x, y]$ denote the interval $\{x, \dots, y\}$.

Furthermore, S is a *circular interval* if it is either an interval or a 2-interval containing both 1 and N. S is a *circular t-interval* if it is the union of t circular intervals. For $x < y$, let $[y, x]$ denote the circular interval $[y, N] \cup [1, x]$.

For a permutation π we say that π' is a *cyclic shift of* π if we can get π' from π by repeatedly moving the last entry to the first position. We say that two orderings are *equivalent* if we can get one from the other by applying cyclic shifts, or by reversing the order and then applying cyclic shifts.

Observation 7.2.1. *If π_1 and π_2 are equivalent, then π_1 is an $(AB)^k$-free ordering if and only if π_2 is $(AB)^k$-free ordering.*

Note that Observation 7.2.1 does not hold for $(AB)^kA$-free hypergraphs. For a set $S \subseteq \{1, 2, \dots, N\}$, let S^c denote the complement of S, that is $S^c = \{1, 2, \dots, N\} \setminus S$. Note that the complement of a circular interval is also a circular interval.

Lemma 7.1. Let $V = \{1, 2, \dots, N\}$ and $\mathcal{E}$ be any collection of subsets of V. Assume that $\mathcal{E}$ contains S and also S^c for some $S \subseteq V$.

a) If π is an $ABAB$-free ordering of $\mathcal{H} = (V, \mathcal{E})$, then either the elements of S or the elements of S^c are consecutive in π. (That is, S and S^c are circular intervals with respect to the ordering π.)

b) If the vertices of S or the elements of S^c are consecutive in π, then neither S nor S^c forms an $ABAB$ pattern with any subset of V.

Proof. For part a), note that if neither the elements of S nor the elements of S^c are consecutive, then S and S^c together form an $ABAB$ pattern.

For part b), we can use Observation 7.2.1 and move S (or S^c) to the beginning of the ordering. If we have an interval at the beginning of the order, it clearly cannot form an $ABAB$ pattern with any other subset of V. ∎

Lemma 7.2. Let $V = \{1, 2, \dots, N\}$, where $N \geq 4$ and let $\mathcal{E}$ be the set of all circular intervals of $[N]$. Then the hypergraph $\mathcal{H} = (V, \mathcal{E})$ with an ordering π is $ABAB$-free if and only if π is equivalent to the increasing order $1, 2, \dots, N$.

Proof. First, consider the order $1, 2, \dots, N$ and a circular interval S. If $x, y \in S$ and $x < y$, then either $[x, y] \subseteq S$ or $[1, x] \cup [y, N] \subseteq S$, hence S is not part of any $ABAB$ pattern. As this holds for any S, the ordering is $ABAB$-free.

In the other direction, let π be an ordering of V not equivalent to $1, 2, \dots, N$. Since $N \geq 4$, we can find $x, y, z, q \in V$ such that $x < y < z < q$, but in π their cyclic order is not x, y, z, q nor q, z, y, x.

- If the cyclic order in π is x, y, q, z or x, z, q, y, then $[q, x]$ and $[y, z]$ are disjoint circular intervals forming an $ABAB$ pattern.
- If the cyclic order in π is x, z, y, q or x, q, y, z, then $[x, y]$ and $[z, q]$ are disjoint circular intervals forming an $ABAB$ pattern.

Hence, any $ABAB$-free ordering is equivalent to $1, 2, \ldots, N$. ■

Let $V = \cup_{i=1}^{k} A_i$ for some disjoint $A_i \subseteq V$. We say that an ordering of V *has structure* $A_1, \ldots, A_k$ if for any $i \neq j$ and $v \in A_i, w \in A_j$, the element v comes before w if and only if $i < j$. That is, other than the internal order within the A_i-s, the ordering is fixed.

Observation 7.2.2. *An ordering π has structure $A_1, \ldots, A_k$ if and only if for any selection $a_i \in A_i$ the a_i-s are in order within π.*

Let $\mathcal{I}_k$ denote the collection of all circular intervals on $\{1, 2, \ldots, k\}$. From Lemma 7.2 and Observation 7.2.2 we obtain the following.

Corollary 7.4. Let $V = \cup_{i=1}^{k} A_i$ for some disjoint $A_i \subseteq V$ and let $\mathcal{E} = \{\bigcup_{i \in \alpha} A_i : \alpha \in \mathcal{I}_k\}$. Then an ordering π of V is $ABAB$-free if and only if π is equivalent to an ordering that has structure $A_1, \ldots, A_k$.

Now we are ready to prove Theorem 7.3.

Proof. Deciding whether a 3-uniform hypergraph has a proper 2-coloring is NP-hard [DRS02]. We reduce this problem to deciding $ABAB$-freeness, thus showing the theorem.

Let $\mathcal{G} = (V_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$ be any given 3-uniform hypergraph, where $V_{\mathcal{G}} = \{v_1, v_2, \ldots, v_n\}$ and $\mathcal{E}_{\mathcal{G}} = \{e_1, e_2, \ldots, e_m\}$. We construct a hypergraph $\mathcal{H} = (V, \mathcal{E})$ such that $\mathcal{G}$ is 2-colorable if and only if $\mathcal{H}$ is $ABAB$-free.

The vertex set V of $\mathcal{H}$ is the set of the first N natural numbers constructed as follows: for each vertex of $\mathcal{G}$ we take two vertices, and for each edge of $\mathcal{G}$ we take four vertices. That is, let $V_1 = \bigcup_{i=1}^{n} R_i$, where $R_i = \{2i-1, 2i\}$ and let $V_2 = \bigcup_{j=1}^{m} S_j$, where $S_j = \{2n + (4j-3), 2n + (4j-2), 2n + (4j-1), 2n + 4j\}$. The vertex set of $\mathcal{H}$ is $V = V_1 \cup V_2$. We will denote by t_j the first vertex of S_j, that is, $t_j = 2n + (4j-3)$ and $S_j = \{t_j, t_j + 1, t_j + 2, t_j + 3\}$.

Now, we will construct the sets of hyperedges in three steps. $\mathcal{E}_1$ is the set of circular intervals of $[N]$ that can be written as the unions of R_i-s and S_j-s. Corollary 7.4 implies that any $ABAB$-free ordering of $(V, \mathcal{E}_1)$ is equivalent to a unique $ABAB$-free ordering that has structure $R_1, \ldots, R_n, S_1, \ldots, S_m$.

Our next goal is to construct a set of edges $\mathcal{E}_2$, such that any $ABAB$-free ordering of $(V, \mathcal{E}_1 \cup \mathcal{E}_2)$ corresponds to a 2-coloring of the vertex set $V_{\mathcal{G}}$, in such a way that the color of a vertex v_i can be determined from the order within R_i, and furthermore, if $v_i \in e_j$, then the color can also be determined from the order within S_j. Take an $ABAB$-free ordering π_1 of $(V, \mathcal{E}_1)$, and let π_2 be the unique $ABAB$-free ordering that is equivalent to π_1 and has structure $R_1, \ldots, R_n, S_1, \ldots, S_m$. We will say that the color of v_i is decided based on the order of $2i-1$ and $2i$ in π_2. We color v_i red if $2i-1$ precedes $2i$ in π_2, and color it blue otherwise.

Consider an edge $e_j = \{v_i, v_k, v_l\}$ with $i < k < l$ and the corresponding set $S_j = \{t_j, t_j+1, t_j+2, t_j+3\}$. Our goal is to ensure that the order of t_j and t_j+1 in π_2 represents the color of v_l, the order of t_j+1 and t_j+2 represents the color of v_k, and the order of t_j+2 and t_j+3 represents the color of v_i. We emphasize that $i < k < l$, that is, the pairs (t_j, t_j+1), (t_j+1, t_j+2), (t_j+2, t_j+3) represent the color of the vertices of the edge in reverse order. For example, the orders $(t_j, t_j+1, t_j+2, t_j+3)$ and $(t_j+3, t_j+2, t_j+1, t_j)$ correspond to the all red and all blue colorings of the vertices in edge e_j, respectively.

To achieve this, for each v_i and edge e_j, where v_i is the r-th smallest vertex in e_j, we add the edges $I_{i,j,r} = \{2i, \ldots, t_j + (3-r)\}$ and $I'_{i,j,r} = \{2i-1, 2i+1, 2i+2, \ldots, t_j + (3-r) - 1, t_j + (3-r) + 1\}$ to $\mathcal{E}_2$. Now $I_{i,j,r}$ and $I'_{i,j,r}$ differ only on 4 elements, $2i-1, 2i, t_j+(3-r), t_j+(3-r)+1$, and the structure forced by $\mathcal{E}_1$ ensures that the first two always come sooner in π_2 than the second two. Edges $I_{i,j,r}$ and $I'_{i,j,r}$ enforce that in any $ABAB$-free ordering $2i-1$ precedes $2i$ if and only if $t_j+(3-r)$ precedes $t_j+(3-r)+1$. See Table 7.1 for the six edges corresponding to a fixed $e_j = \{v_i, v_k, v_l\}$.

	R_i		$\ldots$	R_k		$\ldots$	R_l		$\ldots$	S_j			
$I_{i,j,1}$		**x**	x	x	x	x	x	x	x	x	x	**x**	
$I'_{i,j,1}$	**x**		x	x	x	x	x	x	x	x	x		**x**
$I_{k,j,2}$					**x**	x	x	x	x	x	**x**		
$I'_{k,j,2}$				**x**		x	x	x	x	x		**x**	
$I_{l,j,3}$								**x**	x	**x**			
$I'_{l,j,3}$							**x**		x		**x**		

Table 7.1: Six edges of $\mathcal{H}$ in $\mathcal{E}_2$ corresponding to the edge $e_j = \{v_i, v_k, v_l\}$ of $\mathcal{G}$.

So any $ABAB$-free ordering of $(V, \mathcal{E}_1 \cup \mathcal{E}_2)$ corresponds to a unique coloring of $\mathcal{G}$, but the coloring might not be proper. For later use, we note the following.

Observation 7.2.3. *Let $E \in \mathcal{E}_1 \cup \mathcal{E}_2$ and let D_1 be the first and D_2 the last of the sets $R_1, \ldots, R_n, S_1, \ldots, S_m$ that intersects E. Then the sets in the list $R_1, \ldots, R_n, S_1, \ldots, S_m$ between D_1 and D_2 are subsets of E.*

Now, we add some edges $\mathcal{E}_3$ to ensure that each $ABAB$-free ordering corresponds to a proper 2-coloring. For each $S_j = \{t_j, t_j+1, t_j+2, t_j+3\}$ add $\{t_j, t_j+3\}$ and $\{t_j, t_j+3\}^c$ to $\mathcal{E}_3$. By Lemma 7.1, the possible orderings of S_j in an $ABAB$-free ordering of $(V, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3)$ are the ones where t_j and t_j+3 are neighbors. These do not include the orders that correspond to monochromatic edges, namely the orders $(t_j, t_j+1, t_j+2, t_j+3)$ and $(t_j+3, t_j+2, t_j+1, t_j)$. Hence, each $ABAB$-free ordering of $(V, \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3)$ corresponds to a proper coloring.

We are left with one task: we have to show that for each proper 2-coloring there is an $ABAB$-free ordering. We take the vertices in increasing order, and then within each R_i and S_i, we reorder the vertices. This way the resulting ordering π will automatically have structure $R_1, \ldots, R_n, S_1, \ldots, S_m$. Order an R_i as $2i-1, 2i$ if v_i is red and as $2i, 2i-1$ if v_i is blue. Order an $S_j = \{t_j, t_j+1, t_j+2, t_j+3\}$ that corresponds to $e_j = \{v_i, v_k, v_l\}$ with $i < k < l$ based on the colors of v_i, v_k, v_l according to Table 7.2.

colors of v_i, v_k, v_l	order in S_j
red, red, blue	t_j+1, t_j+2, t_j+3, t_j
red, blue, red	t_j+2, t_j+3, t_j, t_j+1
red, blue, blue	$t_j+2, t_j+1, t_j, t_j+3,$
blue, red, red	t_j+3, t_j, t_j+1, t_j+2
blue, red, blue	t_j+1, t_j, t_j+3, t_j+2
blue, blue, red	$t_j, t_j+3, t_j+2, t_j+1,$

Table 7.2: Ordering rule for the S_j-s.

We can see in Table 7.2 that in each case the pairs (t_j, t_j+1), (t_j+1, t_j+2) and (t_j+2, t_j+3) represent the colors of v_l, v_k, and v_i correctly. Hence, this ordering indeed represents the desired coloring.

Now we show that π is indeed an $ABAB$-free ordering. Since the ordering has the structure $R_1, \ldots, R_n, S_1, \ldots, S_m$, part b) of Lemma 7.1 tells us that the edges of $\mathcal{E}_1$ cannot take part in an $ABAB$ pattern. Similarly, as in each $S_j = \{t_j, t_j+1, t_j+2, t_j+3\}$ we placed t_j adjacent to t_j+3, the edges in $\mathcal{E}_3$ cannot form an $ABAB$ pattern with any edge.

Hence, we only have to check whether two edges from $\mathcal{E}_2$ form an ABAB pattern. First, consider the case when we have two edges corresponding to the same $e_j = \{v_i, v_k, v_l\}$, that is, two rows from Table 7.1. The edge pairs ($I_{i,j,1}$, $I'_{i,j,1}$), ($I_{k,j,2}$, $I'_{k,j,2}$), ($I_{l,j,3}$, $I'_{l,j,3}$) will not form $ABAB$ patterns since the ordering represents the coloring correctly. Further, let $H \in \{I_{i,j,1}, I'_{i,j,1}\}$, $K \in \{I_{k,j,2}, I'_{k,j,2}\}$, and $L \in \{I_{l,j,3}, I'_{l,j,3}\}$. Then $H \supset K \supset L$. Thus, no two edges from $\{I_{i,j,1}, I'_{i,j,1}, I_{k,j,2}, I'_{k,j,2}, I_{l,j,3}, I'_{l,j,3}\}$ can form an $ABAB$ pattern.

Secondly, suppose $H \in \{I_{i_1,j_1,r_1}, I'_{i_1,j_1,r_1}\}$ and $L \in \{I_{i_2,j_2,r_2}, I'_{i_2,j_2,r_2}\}$, where $j_1 < j_2$ and suppose that H and L form an $ABAB$ pattern on vertices $X = (x_1, x_2, x_3, x_4)$. The vertices between $R_{max(i_1,i_2)}$ and S_{j_1} belong to both H and L, hence they cannot be in X. Since $j_1 < j_2$, the vertices of S_{j_1} and later vertices cannot be in $H \setminus L$, so we have at most one vertex of X there. Now consider three cases. If $i_1 = i_2$ we have only two remaining vertices that we can put in X, the vertices $2i_1 - 1$ and $2i_1$, but two vertices are not enough. If $i_1 < i_2$, then the vertices in R_{i_2} and before that cannot belong to $L \setminus H$, so we have at most one vertex of X there, which is not enough. Similarly, if $i_2 < i_1$, the vertices in R_{i_1} and before that cannot belong to $H \setminus L$, so we have at most one vertex of X there, which is not enough. Therefore, the ordering π is $ABAB$-free, and the proof is complete. ■

§ 7.3 $ABABA$-freeness is NP-complete

In this section, we show the following special case of Theorem 7.2.

Theorem 7.5. It is NP-hard to decide if a given hypergraph is $ABABA$-free.

We start with a lemma similar to Lemma 7.2. Note that Observation 7.2.1 does not hold in the $ABABA$-free setting, but an ordering is $ABABA$-free if and only if the reverse of the ordering is $ABABA$-free.

Lemma 7.3. Let $V = \{1, 2, \ldots, N\}$, where $N \geq 7$ and let $\mathcal{E}$ be the set of all 2-intervals of V. Then the hypergraph $\mathcal{H} = (V, \mathcal{E})$ is $ABABA$-free if and only if the ordering of V is $1, 2, \ldots, N$ or $N, N-1, \ldots, 1$.

Proof. It is easy to see that the orderings $1, 2, \ldots, N$ and $N, N-1, \ldots, 1$ are indeed $ABABA$-free. Let π be any linear ordering of $[N]$ other than $1, 2, \ldots, N$ and $N, N-1, \ldots, 1$. Then there must be an $x \in [N]$ such that x and $x+1$ are not neighbors in π. Since $N \geq 7$, there is a $y \in [N]$ such that y is not a neighbor of x or $x+1$. As $x, x+1$ and y are not neighbors in π, we can pick numbers z and q that separate them from each other in the order π. Then $\{x, x+1, y\}$ is a 2-interval which forms an $ABABA$ pattern with the 2-interval $\{z, q\}$. ■

Let $\mathcal{J}_k$ denote the collection of all intervals on $\{1, 2, \ldots, k\}$. Analogous to Corollary 7.4, Lemma 7.3 implies the following.

Corollary 7.6. Suppose $k \geq 7$ and $V = \cup_{i=1}^{k} A_i$ for some disjoint $A_i \subseteq V$. Let $\mathcal{E} = \{(\bigcup_{i \in \alpha} A_i) \cup (\bigcup_{j \in \beta} A_j) : \alpha, \beta \in \mathcal{J}_k\}$. Then, an ordering π of V is $ABABA$-free if and only if π has structure $A_1, A_2, \ldots, A_k$ or $A_k, A_{k-1}, \ldots, A_1$.

The proof of Theorem 7.5 follows the same ideas as the proof of Theorem 7.3. There might be an easy reduction directly from $ABAB$-freeness, but we are not aware of it and give an explicit proof below.

Proof. Once again, we will use that finding 2-colorings for 3-uniform hypergraphs is NP-hard. Let $\mathcal{G}$ be a 3-uniform hypergraph and let $\mathcal{H} = (V, \mathcal{E})$ be constructed the same way as in the proof of Theorem 7.3. As we have seen, $\mathcal{G}$ is 2-colorable if and only if $\mathcal{H}$ is $ABAB$-free. We will construct a hypergraph $\mathcal{H}'$ such that $\mathcal{H}$ is $ABAB$-free if and only if $\mathcal{H}'$ is $ABABA$-free.

Recall that the vertex set V of $\mathcal{H}$ is $[N]$ and $V = (\bigcup_{i=1}^{n} R_i) \cup (\bigcup_{j=1}^{m} S_j)$. We will build $\mathcal{H}'$ on the vertex set $[N+1]$. Let $\mathcal{I}$ be the collection of intervals on $[N+1]$ that are unions of some of the sets from $R_1, \dots, R_n, S_1, \dots, S_m, \{N+1\}$. Let $\mathcal{E}_0$ be the 2-intervals formed from intervals in $\mathcal{I}$, that is $\mathcal{E}_0 = \{I \cup J : \ I, J \in \mathcal{I}\}$. Note that by Corollary 7.6, if the edge set of a hypergraph on $[N+1]$ contains $\mathcal{E}_0$, then any $ABABA$-free ordering will have structure $R_1, \dots, R_n, S_1, \dots, S_m, \{N+1\}$ or $\{N+1\}, S_m, \dots, S_1, R_n, \dots, R_1$.

Consider the hypergraph $\mathcal{H}' = (V', \mathcal{E}')$, where $V' = [N+1]$ and $\mathcal{E}' = \mathcal{E} \cup \mathcal{E}_+ \cup \mathcal{E}_0$ with $\mathcal{E}_+ = \{E \cup \{N+1\} : \ E \in \mathcal{E}\}$. We need to show that $\mathcal{H}$ is $ABAB$-free if and only if $\mathcal{H}'$ is $ABABA$-free. First, let π be any $ABAB$-free ordering of $[N]$ for $\mathcal{H}$. Recall that for any $ABAB$-free ordering π of $\mathcal{H}$, there is an equivalent $ABAB$-free ordering π_2 that has structure $R_1, \dots, R_n, S_1, \dots, S_m$. We will show that the ordering π' that we get by placing $N+1$ after π_2 is an $ABABA$-free ordering for $\mathcal{H}'$.

As no two edges from $\mathcal{E}$ form an $ABAB$ pattern and we introduced only a single new vertex, no two edges from $\mathcal{E} \cup \mathcal{E}_+$ can form an $ABABA$ pattern. Hence, if H and L form an $ABABA$ pattern, we can assume that $H \in \mathcal{E}_0$. Then, as π' has structure $R_1, \dots, R_n, S_1, \dots, S_m, \{N+1\}$, the edge H is not only a 2-interval with respect to the usual ordering, but also with respect to π'. Hence, the only possibility is that H plays the role of B in the pattern. This also implies that L is from $\mathcal{E} \cup \mathcal{E}_+$.

Let D_1 be the first and D_2 the last of the sets $R_1, \dots, R_n, S_1, \dots, S_m, \{N+1\}$ that intersects L. Since H and L form an $ABABA$ pattern and H is a 2-interval, H must lie between D_1 and D_2. Recall that $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3$ and from Observation 7.2.3 if $L \in \mathcal{E}_1 \cup \mathcal{E}_2$, then the sets in the list $R_1, \dots, R_n, S_1, \dots, S_m$ between D_1 and D_2 are subsets of L. This would imply that $H \subseteq L$, a contradiction. So we must have $L \in \mathcal{E}_3 \cup \mathcal{E}_+$. From Lemma 7.1 we know that an edge from $\mathcal{E}_3$ cannot form an $ABAB$-pattern with anything. So, let $L \in \mathcal{E}_+$. Since L forms an $ABABA$ pattern with H, the edge $L' = L \setminus \{N+1\} \in \mathcal{E}$ must form an $ABAB$ pattern with H; a contradiction by the previous arguments. Hence π' is $ABABA$-free for H'.

Conversely, let $\mathcal{H}'$ be $ABABA$-free and π' be any $ABABA$-free ordering for $\mathcal{H}'$. By Lemma 7.3, $N+1$ should be at the last or the first position in π'. Assume the former, without loss of generality. We claim that the ordering π that we get by deleting $N+1$ from π' is an $ABAB$-free ordering for $\mathcal{H}$. Indeed, if there is an $ABAB$ pattern for two hyperedges H, L of $\mathcal{H}$ in π, then H, L are also hyperedges of $\mathcal{H}'$ and they form an $ABAB$ pattern in π'. Assume, without loss of generality, that H plays the role of A in the $ABAB$ pattern. This implies that there is an $ABABA$ pattern formed by $H_+ = H \cup \{N+1\}$ and L in π' for the hypergraph $\mathcal{H}'$; a contradiction. Hence, $\mathcal{H}$ is $ABAB$-free. This completes the proof. ∎

§ 7.4 Generalizing to $(AB)^k$ and $(AB)^k A$

We show Theorems 7.1 and 7.2 using induction on k. The base cases are covered by Theorems 7.3 and 7.5.

For a hypergraph $\mathcal{H} = (V, \mathcal{E})$ and $t \in \mathbb{N}$, let $\mathcal{H}_t$ denote the hypergraph whose vertex set is $V' = V \cup X$ for some new vertices $X = \{x_1, x_2, \ldots, x_t\}$ and the edge set is $\mathcal{E}' = \mathcal{E} \cup \{E \cup \{x\} : E \in \mathcal{E},\ x \in X\}$.

Lemma 7.4. Let $\mathcal{H} = (V, \mathcal{E})$ be a given hypergraph. Then for any $k \in \mathbb{N}$ the hypergraph $\mathcal{H}$ is $(AB)^{k-1}A$-free if and only if $\mathcal{H}_{2k+1}$ is $(AB)^k A$-free.

Proof. Let $\mathcal{H}$ be $(AB)^{k-1}A$-free and π be an $(AB)^{k-1}A$-free ordering of V. We will show that $\mathcal{H}_{2k+1} = (V', \mathcal{E}')$ is $(AB)^k A$-free. Let $\pi = v_1, v_2, \ldots, v_n$. Consider the ordering $\pi' = v_1, \ldots, v_n, x_1, \ldots, x_{2k+1}$ of V'. If there are hyperedges $H', L' \in \mathcal{E}'$ which form an $(AB)^k A$ pattern, then there are vertices $h_1, l_1, \ldots, h_k, l_k, h_{k+1}$ in this order in π' such that $h_i \in H' \setminus L'$ for $1 \le i \le k+1$ and $l_j \in L' \setminus H'$ for $1 \le j \le k$. Let $H, L \in \mathcal{E}$ be the intersections of H', L' respectively, with the vertex set V. Since H' and L' contain at most one vertex of X, we have $|H' \setminus H| \le 1$ and $|L' \setminus L| \le 1$. Also, as $\mathcal{H}$ is $(AB)^{k-1}A$-free, in the sequence $h_1, l_1, \ldots, h_k, l_k, h_{k+1}$, at least the last three vertices must be from X otherwise H and L form an $(AB)^{k-1}A$ pattern in the ordering π. However, this is not possible because of $|H' \setminus H| \le 1$ and $|L' \setminus L| \le 1$. Thus π' is an $(AB)^k A$-free ordering for $\mathcal{H}_{2k+1}$.

Conversely, assume that $\mathcal{H}_{2k+1}$ is $(AB)^k A$-free. We will show that $\mathcal{H}$ is $(AB)^{k-1}A$-free. Let π' be an $(AB)^k A$-free ordering of V' for the hypergraph $\mathcal{H}_{2k+1}$ and let π be an induced ordering of π' on the vertex set V. Suppose there are hyperedges $H, L \in \mathcal{E}$ which form an $(AB)^{k-1}A$ pattern in $\mathcal{H}$ and let $h_1, l_1, \ldots, h_{k-1}, l_{k-1}, h_k$ be vertices in this order in π such that $h_i \in H \setminus L$ for $1 \le i \le k$ and $l_j \in L \setminus H$ for $1 \le j \le k-1$. Since, $|X| = 2k+1$ and $|\{h_1, l_1, \ldots, h_{k-1}, l_{k-1}, h_k\}| = 2k-1$, it follows that in the ordering π', there must be at least two vertices $x, y \in X$ which lie between two consecutive elements of the sequence $h_1, l_1, \ldots, h_{k-1}, l_{k-1}, h_k$ or they both occur before or after all the elements of this sequence. Then either $H \cup \{x\}$ and $L \cup \{y\}$, or $H \cup \{y\}$ and $L \cup \{x\}$ form an $(AB)^k A$ pattern in π'; a contradiction. Hence, π is an $(AB)^{k-1}A$-free ordering for $\mathcal{H}$. ∎

Lemma 7.5. Let $\mathcal{H} = (V, \mathcal{E})$ be a given hypergraph. Then for any $k \in \mathbb{N}$ the hypergraph $\mathcal{H}$ is $(AB)^k$-free if and only if $\mathcal{H}_{2k+2}$ is $(AB)^{k+1}$-free.

Proof. The proof is the same as that of Lemma 7.4. ∎

Now the proofs of Theorems 7.1 and 7.2 are direct implications of the two lemmas above, and Theorems 7.3 and 7.5.

§ 7.5 The Complexity of Recognition by Pseudodisks

A family of *pseudocircles* is a set of closed Jordan curves such that every two of them are either disjoint, intersect at exactly one point in which they touch, or intersect at exactly two points in which they properly cross each other. A family of *pseudodisks* is a collection of compact planar regions whose boundaries form a family of pseudocircles.

A family $\mathcal{A}$ of pseudodisks in the plane is called a *stabbed* pseudodisk arrangement if there is a point in the plane that lies in every pseudodisk of $\mathcal{A}$.

Theorem 7.7 (Ackerman, Keszegh, Pálvölgyi [AKP20])**.** A hypergraph can be realized as the incidence hypergraph of stabbed pseudodisks and points in the plane if and only if it is $ABAB$-free.

Theorem 7.8. It is NP-complete to decide whether an abstract hypergraph can be realized as the incidence hypergraph of points and pseudodisks in the plane or not.

Proof. The problem is in NP, because we can give a short combinatorial description of the realizing arrangement.

To prove NP-hardness, we show that $ABAB$-freeness is reducible to recognizing pseudodisk hypergraphs. Let $\mathcal{H} = (V, \mathcal{E})$ be a given hypergraph, and we want to decide whether it is $ABAB$-free. Consider the hypergraph $\mathcal{H}' = (V', \mathcal{E}')$, where $V' = V \cup \{x\}$ and $\mathcal{E}' = \{E \cup \{x\} : E \in \mathcal{E}\}$. Clearly, $\mathcal{H}'$ can be represented as a hypergraph of pseudodisks in the plane if and only if $\mathcal{H}'$ can be represented as a hypergraph of *stabbed* pseudodisks in the plane. By Theorem 7.7, a hypergraph is a hypergraph of stabbed pseudodisks in the plane if and only if it is $ABAB$-free. Furthermore, as x lies in every edge, it cannot take part in an $ABAB$ pattern. Hence H' is $ABAB$-free if and only if H is $ABAB$-free. Therefore, H is $ABAB$-free if and only if H' can be represented as the incidence hypergraph of points and pseudodisks in the plane.

Applying Theorem 7.3 we conclude that it is NP-hard to decide if $\mathcal{H}$ is a hypergraph of points and pseudodisks in the plane. ■

§ 7.6 Concluding Remarks and Open Questions

We studied vertex ordering problems on hypergraphs that forbid a fixed pattern of a sequence, and we showed that it is NP-hard to decide if a hypergraph is $(AB)^k$-free or $(AB)^kA$-free for any $k \geq 2$. In relation to the recognition by geometric objects, this also implies the NP-hardness of deciding if a hypergraph can be recognized by t-intersecting upward curves for any $t \geq 2$.

Further, in the proofs, we have used k-intervals often, which is strongly connected to the so-called consecutive ones property. Let us say that a $(0,1)$-matrix has the k-*consecutive ones property* if there exists a row order such that in each column the occurrences of all ones appear in at most k consecutive blocks. Clearly, the incidence matrix of a hypergraph has the k-consecutive ones property if and only if there is an ordering of the vertices such that every hyperedge is a union of at most k intervals in that ordering.

Deciding if a given (0,1)-matrix has the 1-consecutive ones property can be done in polynomial time [BL76; FG65]. On the other hand, for $k \geq 2$ it is NP-complete to decide whether a given (0,1)-matrix has the k-consecutive ones property [Gol+95]. This latter result would also follow from our methods with simple modifications, even for the circular k-interval case, but we omit these.

▷ Limitations.

Our contribution addresses the problem almost completely, with a few restricted settings left open. Primarily, the most natural question is about the missing $k = 1$ case in Theorem 7.2. Secondly, in our proofs of Theorems 7.3 and 7.5, there are some hyperedges that have a linear number of vertices. As mentioned in the introduction, the $ABAB$-freeness of a 2-uniform hypergraph can be decided in polynomial time since they are exactly the outerplanar graphs [AKP20]. However, we do not know if the problem is in class P even for 3-uniform hypergraphs; maybe these problems become tractable if we bound the sizes of the hyperedges. These open questions are formally stated below.

> **Open Problem 14.** Is it NP-complete to decide if a given hypergraph is ABA-free?

Open Problem 15. What is the complexity of deciding if a c-uniform hypergraph is $(AB)^k$-free or $(AB)^k A$-free for an absolute constant c, and $k \geq 2$?

To close this chapter, we briefly discuss results related to ABA-freeness. We mention some related results. It was proved by Opatrny [Opa79] that the related BETWEENNESS problem is NP-complete. Here, our input is a collection of ordered triples of some base set V, and the question is whether there is an ordering of V in which for all ordered triples, the middle element is between the other two elements of the triple in the ordering. It is not hard to see that this implies that NON-BETWEENNESS is also NP-complete, i.e., when our input is a collection of ordered triples of some base set V, and the question is whether there is an ordering of V for which in all ordered triples the middle element is *not* in the middle. Deciding ABA-freeness is a special case of NON-BETWEENNESS.

Chapter 8

Conclusion

In this chapter, we summarize the work done in this thesis and list some open problems. Most of these open problems listed in this chapter also appeared in the previous chapter, but we include them here for completeness.

This thesis mainly delves into the structural and algorithmic aspects of hypergraph support - a graph Q on the vertices of the hypergraph such that the vertices of each hyperedge induce a connected subgraph of Q. Previous work shows that it is NP-hard to decide if a hypergraph has a support that is a planar or even a 2-outerplanar graph [JP87; Buc+11]. There are restricted settings for geometric hypergraphs which admit planar support, like hypergraphs that are defined by non-piercing regions in the plane [RR20]. We extended these results to a more general class of hypergraphs by introducing a novel framework where a hypergraph is defined by a host graph G and its connected subgraphs satisfying some additional properties.

Table 8.1 below summarizes our results on the construction of support for primal, dual, and intersection hypergraphs, and the running time of computation for different graph classes. In the table, we make use of the following notations:
- $\mathcal{G}_g$: the class of all graphs of genus at most g.
- $\mathcal{C}_o$: the class of all the outerplanar graphs.
- $\mathcal{F}_t$: the class of all the graphs of treewidth at most t.
- If the hypergraph is defined by a host graph G, and collections $\mathcal{H}$ and $\mathcal{K}$ of connected subgraphs of G, then $n = |V(G)|$, and $m = \max\{|\mathcal{H}|, |\mathcal{K}|\}$.

In the light of Table 8.1, we propose the following open problems.

Open Problem 16. Let G be an n-vertex graph of bounded genus and $c : V(G) :\rightarrow \{\mathbf{b}, \boldsymbol{r}\}$ be a 2-coloring of the vertices of G. Let $\mathcal{H}$ and $\mathcal{K}$ be two families of sub-

Host graph	Subgraphs	Primal	Dual	Intersection
$\mathcal{G}_g$	cross-free	$\mathcal{G}_g$ open	$\mathcal{G}_g$ open	$\mathcal{G}_g$ open
$\mathcal{C}_o$	non-piercing	$\mathcal{C}_o$ $r.t.\ O(n^6)$	$\mathcal{C}_o$ $r.t.\ O(n^6)$	$\mathcal{C}_o$ $r.t.\ O(n^6)$
$\mathcal{F}_t$	non-piercing	$\mathcal{F}_{O(2^t)}$ $r.t.\ \text{poly}(n^t)$	$\mathcal{F}_{O(2^{4t})}$ $r.t.\ \text{poly}(n^t)$	$\mathcal{F}_{O(2^{2^t})}$ $r.t.\ \text{poly}(n^{2^t})$

Table 8.1: Supports for primal, dual and intersection hypergraphs under various settings. In the table, "$r.t.$" refers to the "running time" of computing the corresponding support.

graphs of G with $|\mathcal{H}| = m_1, |\mathcal{K}| = m_2$ such that $(G, \mathcal{H})$ and $(G, \mathcal{K})$ are simultaneously cross-free in some embedding of G. Do the algorithms to compute primal, dual, or intersection supports, as established in Theorems 2.3 to 2.5, run in time $\text{poly}(n, m_1, m_2)$? If not, are there any other efficient algorithms to construct supports of bounded genus?

Table 8.1 shows that cross-free is a sufficient condition to construct an intersection support of bounded genus. However, the result fails even for the dual case if cross-free is replaced by non-piercing; see the construction in Example 2.1. We are not aware of any such construction for the primal setting.

Open Problem 17. Give a non-piercing graph system $(G, \mathcal{H})$ of genus g with a 2-coloring $c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$. Does a primal support of genus g exist?

Further, we could not provide an exact characterization for the existence of a bounded genus support, and thus state the following as an open problem.

Open Problem 18. For a connected system $(G, \mathcal{H}, \mathcal{K})$ of genus g, cross-free is a sufficient condition for the existence of an intersection (and hence primal and dual) support of genus at most g. What are the characterizations for the existence of a primal, dual or intersection support that has genus at most g?

Buchin et al. [Buc+11] left open the complexity of deciding the existence of an outerplanar support. By Theorem 3.8, if an abstract hypergraph is $abab$-free, it admits an outerplanar support. However, the converse is not true. The following question may be of independent interest.

Open Problem 19. What are the necessary and sufficient conditions for the existence of an outerplanar support for $\mathcal{H}$?

Open Problem 20. Since $abab$-freeness is a sufficient condition for the existence of an outerplanar support, what are the sufficient conditions for the existence of a support that is a 2-outerplanar graph, or more generally, a k-outerplanar graph?

Open Problem 21. If $(G, \mathcal{H}, \mathcal{K})$ is an outerplanar non-piercing intersection system, there is an intersection support that is outerplanar (Theorem 3.3). If G is k-outerplanar, does the non-piercing system $(G, \mathcal{H}, \mathcal{K})$ admit an intersection support that is $c(k)$-outerplanar, where $c(k)$ depends only on k.

We proved in Chapter 4 that if $(G, \mathcal{H})$ is a non-piercing graph system of treewidth t, then there is a primal support of treewidth $O(2^t)$ and a dual support of treewidth $O(2^{4t})$. We also gave constructions on the lower bounds of the treewidth of primal and dual supports. The following question asks to reduce the gaps between the lower and upper bounds.

Open Problem 22. For non-piercing graph systems $(G, \mathcal{H})$ of treewidth t, there are gaps between the lower and upper bounds on the treewidth of both primal and dual supports as shown in Theorems 4.2, 4.3, 4.7 and 4.8. This motivates further investigation into whether the current gaps between the bounds can be narrowed or eliminated through improved constructions.

We know that if $(G, \mathcal{H}, \mathcal{K})$ is a non-piercing system of treewidth t, it admits an intersection support of treewidth $2^{O(2^t)}$. However, for the primal and dual settings, there are supports that are single exponential in t. This motivates the following interesting open question.

Open Problem 23. We know that if $(G, \mathcal{H}, \mathcal{K})$ is a non-piercing graph system of treewidth t, it admits an intersection support of treewidth $2^{O(2^t)}$ (Theorem 4.4). Is there an intersection support that is a single exponential in t? Or, is there a construction similar to the primal and dual settings shown in Theorems 4.7 and 4.8 that implies a double exponent on the treewidth of an intersection support?

Outerplanar graphs have treewidth at most 2. A non-piercing outerplanar system $(G, \mathcal{H}, \mathcal{K})$ has an outerplanar intersection support. We saw in Theorems 4.7 and 4.8 that for a non-piercing graph system of treewidth t, a primal and dual support can have treewidth exponential in t. We wonder if for a non-piercing graph system of treewidth 2, there exists a primal, dual and intersection support of treewidth at most 2. Recall that a graph is series-parallel iff it does not contain a subgraph homeomorphic to K_4.

Open Problem 24. Given a non-piercing system $(G, \mathcal{H}, \mathcal{K})$. Does there exist a 2-outerplanar support if G has treewidth 2? Does the result hold for *series parallel graphs*?

For an abstract hypergraph $\mathcal{H}$, the results in [JP87; Buc+11] show NP-hardness of deciding if $\mathcal{H}$ has a support that is a planar or a 2-outerplanar graph. We ask a similar question about the graph classes studied in this thesis.

Open Problem 25. Given an abstract hypergraph $\mathcal{H}$, what is the complexity of deciding if $\mathcal{H}$ admits a support that is (i) outerplanar, or (ii) has genus $\leq g$, or (iii) has treewidth $\leq t$?.

Let's move on to geometric hypergraphs. In Chapter 6, we saw applications of sparse support for the packing, covering, and coloring problems for hypergraphs defined by graph systems, as well as those defined by geometric objects. For geometric hypergraphs, we extended the result about planar support for non-piercing regions in $\mathbb{R}^2$, to higher genus surfaces, with the restriction that the regions should be simply connected and weakly non-piercing. In general, dropping either of the two assumptions leads to hardness results for packing and covering problems discussed in previous chapters and hence, does not allow the existence of a sparse support. This motivates the following problem.

Open Problem 26. For hypergraphs satisfying only one of the conditions – (weakly) non-piercing and simply connectedness, what further restrictions should be imposed so as to obtain a sparse support and hence PTAS for the packing and covering problems for the resulting hypergraphs?

Any hypergraph $\mathcal{H}$ satisfies $\chi(\mathcal{H}) \leq \chi(Q)$ for any weak support Q. Unfortunately, this is a very loose bound - the hypergraphs considered in Examples 2.2 and 2.3 are 2-colorable, but any weak support is a complete graph. It would be an interesting problem to investigate the class of hypergraphs for which this bound is tight. We state this as an open problem below.

Open Problem 27. Characterize the hereditary family $\mathcal{F}$ of hypergraphs such that for each hypergraph $\mathcal{H} \in \mathcal{F}$, $\chi(\mathcal{H}) = \Theta(\chi(Q))$ for some appropriate weak support

Q of $\mathcal{H}$.

Hypergraphs defined by planar non-piercing regions, or those defined by weakly non-piercing simply connected regions on surfaces with constant genus, satisfy $\chi(\mathcal{H}) = \Theta(\chi(Q))$ and thus, $\mathcal{F} \neq \emptyset$. Extending this line of investigation to broader classes of geometric hypergraphs represents a natural direction for further research.

Next, we turn to a special case of non-piercing regions, namely, axis-aligned non-piercing rectangles. In Chapter 5, we considered hypergraphs defined by these objects in $\mathbb{R}^2$, and we presented a simple algorithm that computes a straight line drawing of the planar support. The algorithm runs in time $O(n \log^2 n + (n+m) \log m)$, and significantly improves the running time over the previous work of Raman and Ray [RR20]. We conjecture that this bound can be further refined.

Open Problem 28. Given a hypergraph $(P, \mathcal{R})$ defined by a set P of n points and a set $\mathcal{R}$ of m axis-aligned non-piercing rectangles in $\mathbb{R}^2$. Is it possible to reduce the running time of Algorithm 1 to $(n+m)\log(n+m)$?

We suggest a few questions motivated by the work of Raman and Ray [RR20], who obtained polynomial time algorithms to construct *dual* and *intersection* supports for non-piercing regions. In our context, these translate into the following two questions:

Open Problem 29. Given a set of points P and axis-parallel non-piercing rectangles $\mathcal{R}$ in the plane. Similar to Algorithm 1, is there a fast algorithm to construct a plane *dual* support, i.e., a graph $Q^* = (\mathcal{R}, E)$ on $\mathcal{R}$ such that for each point $p \in P$, the rectangles containing p induce a connected subgraph in Q^*.

Open Problem 30. Given two sets $\mathcal{R}$, $\mathcal{S}$, each of which is a collection of axis-parallel non-piercing rectangles in $\mathbb{R}^2$. Is there a fast algorithm to construct a plane *intersection* support, i.e., a graph $\tilde{Q} = (\mathcal{R}, F)$ on $\mathcal{R}$ such that for each $S \in \mathcal{S}$, the induced graph of $\tilde{Q}$ on $\{R \in \mathcal{R} : R \cap S \neq \emptyset\}$ is connected.

Another interesting question is to obtain fast algorithms when, instead of non-piercing axis-parallel rectangles, we have non-piercing convex sets. In this case, it can be shown that a maximal set of edges (straight segments joining pairs of points) which do not discretely pierce any of the convex sets and are pairwise non-crossing, form a planar support. This yields a polynomial-time algorithm, but it is not clear whether a faster algorithm exists that improves the running time over the work done in [RR20].

We now turn to the problems of realizing abstract hypergraphs by vertex orderings of geometric interpretations. Recall that a hypergraph is $ABAB$-free (equivalently, has a cyclic $abab$-free ordering) if and only if it can be realized by stabbed pseudodisks in $\mathbb{R}^2$ [AKP20]. In Chapter 7, we showed the NP-hardness of deciding if a hypergraph is $ABAB$-free. These hypergraphs admit an outerplanar support. It is not difficult to show that every planar graph G (as a hypergraph) admits a representation by pseudodisks in the plane - making two copies of each edge in a planar drawing of G with some topological modification of the edges around each vertex. In particular, the complete graph K_4 can be realized with pseudodisks, but (as a hypergraph) it does not admit an outerplanar support. See Fig. 8.1 for the relation between these classes of hypergraphs. This insight motivates the following natural question.

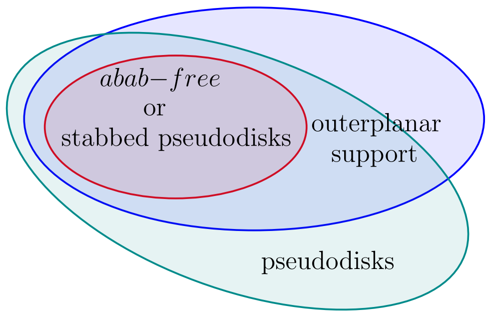


Figure 8.1: Venn diagram showing relation between hypergraph that are (i) $abab$-free, (ii) have outerplanar support, and (iii) have realization with pseudodisks.

> **Open Problem 31.** A hypergraph is $abab$-free if and only if it can be realized by *stabbed* pseudodisks $\mathbb{R}^2$. What are similar geometric characterizations for hypergraphs possessing an outerplanar support?

The NP-hardness result of deciding $ABAB$-freeness was generalized to any alternate sequence of A's and B's of length at least 4 (Theorems 7.1 and 7.2). The only interesting case that of left open is for the sequence of length 3, which we state as an open problem below.

> **Open Problem 32.** What is the complexity of deciding if a hypergraph is ABA-free?

In our proofs for the NP-hardness $ABAB$-free or $ABABA$-free, we used some unbounded edges. We wonder if the problems become tractable if the size of each hyperedge is bounded.

Open Problem 33. What is the complexity of deciding if a *c-uniform* hypergraph is $(AB)^k$-free or $(AB)^k A$-free for an absolute constant c, and a fixed positive integer $k \geq 2$?

▷ Future research directions.

In Part A of the thesis, we gave sufficient conditions on the subgraphs of a host graph for the existence of a support that has bounded genus, bounded treewidth or is an (outer)planar graph. However, for the algorithmic applications, in particular for the existence of PTAS for several packing and covering problems, we need a support that is a sparse graph since it satisfies the properties of a local search graph required for the analysis of local search algorithms. In this vein, we present the following questions, which we believe are of independent interest. Currently, we have no idea how this class of subgraphs look like, and the questions below provide insight into a future research direction.

Question 8.1. Given a graph system $(G, \mathcal{H})$ or an intersection system $(G, \mathcal{H}, \mathcal{K})$. What are the necessary and sufficient conditions to be satisfied by the subgraphs in $\mathcal{H} \cup \mathcal{K}$ to ensure the existence of a (primal, dual or intersection) support that has a linear number of edges, or more generally, a sparse graph, i.e., a support from a hereditary family of graphs with sublinear-sized balanced separators?

Instead of just a sparse support, one may also impose further restrictions on its structure. This is a possible research direction for structural graph theorists. We present one such problem below.

Question 8.2. Given a graph system $(G, \mathcal{H})$ or an intersection system $(G, \mathcal{H}, \mathcal{K})$. What properties should the subgraphs in $\mathcal{H} \cup \mathcal{K}$ satisfy so that the associated hypergraph admits a primal, dual or intersection support that is triangle-free or, more generally, a support that is H-minor free for some fixed graph H?

Epilogue

Chapter I

List of Definitions

Chapter II

List of Figures

CHAPTER III

List of Tables

Chapter IV

List of Notations

Unless otherwise stated, the following symbols/terminology are used in this thesis.

Symbol / Term	**Meaning / Description**
$V(G)$	Vertex set of a graph G.
$E(G)$	Edge set of a graph G.
$G[S]$	Subgraph of G induced on the vertex set $S \subseteq V(G)$.
$v \in G$	Shorthand for $v \in V(G)$.
$e \in G$	Shorthand for $e \in E(G)$.
Host graph	Usually denoted by G.
$\mathcal{H}, \mathcal{K}$	Collections of connected induced subgraphs of a host graph G.
$\mathcal{H}_v$	$\{H \in \mathcal{H} : v \in V(H)\}$.
$\mathcal{H}_e$	$\{H \in \mathcal{H} : e \in E(H)\}$.
Primal hypergraph $(G, \mathcal{H})$	Hypergraph with vertex set $V(G)$, and hyperedges $\{V(H) : H \in \mathcal{H}\}$.
Dual hypergraph $(G, \mathcal{H})$	Hypergraph with vertex set $\mathcal{H}$, and hyperedges $\{\mathcal{H}_v : v \in V(G)\}$.
Intersection hypergraph $(G, \mathcal{H}, \mathcal{K})$	Hypergraph with vertex set $\mathcal{H}$, and hyperedges $\mathcal{H}_K = \{H \in \mathcal{H} : V(H) \cap V(K) \neq \emptyset\}$ for each $K \in \mathcal{K}$.
Support	A graph on the vertex set of a hypergraph such that each hyperedge induces a connected subgraph.

$Q, Q^*, \tilde{Q}$	Primal, dual and intersection supports respectively.
$c : V(G) \to \{\mathbf{b}, \boldsymbol{r}\}$	An arbitrary (not necessarily proper) 2-coloring of the vertices of G, with $\mathbf{b}$ = blue and $\boldsymbol{r}$ = red.
$\mathbf{b}(X), \boldsymbol{r}(X)$	Sets of respectively, blue and red vertices of a subgraph X under the coloring c.

For subgraphs X and Y of G

(i) $X \cap Y$	Subgraph of G induced on $V(X) \cap V(Y)$.
(ii) $X \cup Y$, $X \setminus Y$	Defined analogously.
(iii) $X \subseteq Y$	Means $V(X) \subseteq V(Y)$.

$\subseteq$	Subset or equal.
$\subsetneq$	Subset without equality.
Primal cross-free system $(G, \mathcal{H})$ of genus g	G has genus g, and $\mathcal{H}$ is a collection of cross-free subgraphs of G.
Dual cross-free $(G, \mathcal{H})$, and intersection cross-free $(G, \mathcal{H}, \mathcal{K})$ systems of genus g	Defined analogously.
Primal non-piercing system $(G, \mathcal{H})$ of treewidth t	G has treewidth t, and $\mathcal{H}$ is a collection of non-piercing subgraphs of G.
Dual non-piercing $(G, \mathcal{H})$, and intersection non-piercing $(G, \mathcal{H}, \mathcal{K})$ systems of treewidth t	Defined analogously.
$\rho(R)$	Boundary of region R.

Notations for Chapter 5

$\urcorner$, $\lrcorner$	L-shaped edges between some points in $\mathbb{R}^2$.

$R(p,q)$	Axis-aligned rectangle with diagonally opposite ends p and q.
$h(pq)$	Horizontal segment of the L-shaped edge pq.
$v(pq)$	Vertical segment of the L-shaped edge pq.
ℓ_p	Vertical line through the point p.
$(x(p), y(p))$	x, y coordinates of point p.
$x_-(R), x_+(R)$	Respectively, left and right x-coordinates of rectangle R.
$y_-(R), y_+(R)$	Respectively, lower and upper y-coordinates of rectangle R.
PIECE(R, H)	Sub-rectangle of R intersected by the half plane H.
PIECE(R, p)	Sub-rectangle of R on the left of ℓ_p.
$R[x_-, x_+]$	Sub-rectangle of R with x-coordinates x_- and x_+.
$R[y_-, y_+]$	Sub-rectangle of R with y-coordinates y_- and y_+.
ACTIVE(s)	Set of rectangles *active* at s.
CONTAIN (s,p)	Elements of ACTIVE(s) containing p.
ABOVE(s, p)	Elements of ACTIVE(s) that lie above p.
BELOW(s, p)	Elements of ACTIVE(s) that lie below p.
$\sqcup$	Disjoint union.
UB(p), LB(p)	Respectively, the upper and lower barriers at p.
UPB(R, p), LPB(R, p)	Respectively, the upper and lower piercing barriers at p w.r.t. rectangle R.
SLAB(R, p)	Sub-rectangle of R with x-coordinates $x_-(R)$ & ℓ_p, and y-coordinates $y_+($UPB$(R, p))$ & $y_-($LPB$(R, p))$.

Bibliography

[AH15] Meysam Alishahi and Hossein Hajiabolhassan. "On the chromatic number of general Kneser hypergraphs". In: *Journal of Combinatorial Theory, Series B* 115 (2015), pp. 186–209. ISSN: 0095-8956. DOI: https://doi.org/10.1016/j.jctb.2015.05.010 (cit. on p. 159).

[AHPW19] Anna Adamaszek, Sariel Har-Peled, and Andreas Wiese. "Approximation schemes for independent set and sparse subsets of polygons". In: *Journal of the ACM (JACM)* 66.4 (2019), pp. 1–40 (cit. on p. 9).

[AK00] Paola Alimonti and Viggo Kann. "Some APX-completeness results for cubic graphs". In: *Theoretical Computer Science* 237.1 (2000), pp. 123–134. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(98)00158-3. URL: https://www.sciencedirect.com/science/article/pii/S0304397598001583 (cit. on pp. 147, 149).

[AKP20] Eyal Ackerman, Balázs Keszegh, and Dömötör Pálvölgyi. "Coloring hypergraphs defined by stabbed pseudo-disks and ABAB-free hypergraphs". In: *SIAM Journal on Discrete Mathematics* 34.4 (2020), pp. 2250–2269 (cit. on pp. 30, 31, 44, 78, 79, 151, 152, 159, 168, 169, 176).

[Alo+23] Noga Alon, Allan Grønlund, Søren Fuglede Jørgensen, and Kasper Green Larsen. "Sublinear Time Shortest Path in Expander Graphs". In: *CoRR* abs/2307.06113 (2023). DOI: 10.48550/ARXIV.2307.06113. arXiv: 2307.06113. URL: https://doi.org/10.48550/arXiv.2307.06113 (cit. on p. 9).

[AMM17] Daniel Antunes, Claire Mathieu, and Nabil H. Mustafa. "Combinatorics of Local Search: An Optimal 4-Local Hall's Theorem for Planar Graphs". In: *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*. 2017, 8:1–8:13 (cit. on p. 8).

[Anc+06] Emmanuelle Anceaume, Maria Gradinariu, Ajoy Kumar Datta, Gwendal Simon, and Antonino Virgillito. "A semantic overlay for self-peer-to-peer publish/subscribe". In: *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. IEEE. 2006, pp. 22–22 (cit. on p. 8).

[Aro+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. "Unique games on expanding constraint graphs are easy: extended abstract". In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia,* *Canada, May 17-20, 2008*. Ed. by Cynthia Dwork. ACM, 2008, pp. 21–28. DOI: 10.1145/1374376.1374380. URL: https://doi.org/10.1145/1374376.1374380 (cit. on p. 9).

[Aro97] Sanjeev Arora. "Nearly Linear Time Approximation Schemes for Euclidean TSP and other Geometric Problems". In: *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*. IEEE Computer Society, 1997, pp. 554–563. DOI: 10.1109/SFCS.1997.646145. URL: https://doi.org/10.1109/SFCS.1997.646145 (cit. on p. 9).

[Asc+13] Rom Aschner, Matthew J. Katz, Gila Morgenstern, and Yelena Yuditsky. "Approximation Schemes for Covering and Packing". In: *WALCOM: Algorithms and Computation, 7th International Workshop, WALCOM 2013, Kharagpur, India, February 14-16, 2013. Proceedings*. Ed. by Subir Kumar Ghosh and Takeshi Tokuyama. Vol. 7748. Lecture Notes in Computer Science. Springer, 2013, pp. 89–100. DOI: 10.1007/978-3-642-36065-7_10 (cit. on pp. 11, 132, 134).

[AST90] Noga Alon, Paul Seymour, and Robin Thomas. "A separator theorem for nonplanar graphs". In: *Journal of the American Mathematical Society* 3.4 (1990), pp. 801–808 (cit. on pp. 9, 11, 12, 26).

[Bal+07a] Roberto Baldoni, Roberto Beraldi, Vivien Quema, Leonardo Querzoni, and Sara Tucci-Piergiovanni. "TERA: topic-based event routing for peer-to-peer architectures". In: *Proceedings of the 2007 inaugural international conference on Distributed event-based systems*. 2007, pp. 2–13 (cit. on p. 8).

[Bal+07b] Roberto Baldoni, Roberto Beraldi, Leonardo Querzoni, and Antonino Virgillito. "Efficient publish/subscribe through a self-organizing broker overlay and its application to SIENA". In: *The Computer Journal* 50.4 (2007), pp. 444–459 (cit. on p. 8).

[Ber+11] Sergey Bereg, Minghui Jiang, Boting Yang, and Binhai Zhu. "On the red/blue spanning tree problem". In: *Theoretical computer science* 412.23 (2011), pp. 2459–2467 (cit. on p. 8).

[Ber+15] Sergey Bereg, Krzysztof Fleszar, Philipp Kindermann, Sergey Pupyrev, Joachim Spoerhase, and Alexander Wolff. "Colored non-crossing Euclidean Steiner forest". In: *Algorithms and Computation: 26th International Symposium, ISAAC 2015, Nagoya, Japan, December 9-11, 2015, Proceedings*. Springer. 2015, pp. 429–441 (cit. on p. 8).

[BK96] Hans L. Bodlaender and Ton Kloks. "Efficient and Constructive Algorithms for the Pathwidth and Treewidth of Graphs". In: *Journal of Algorithms* 21.2 (1996), pp. 358–402. ISSN: 0196-6774. DOI: https://doi.org/10.1006/jagm.1996.0049. URL: https://www.sciencedirect.com/science/article/pii/S0196677496900498 (cit. on p. 95).

[BL76] Kellogg S. Booth and George S. Lueker. "Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms". In: *Journal of Computer and System Sciences* 13.3 (1976), pp. 335–379. ISSN: 0022-0000. DOI: https://doi.org/10.1016/S0022-0000(76)80045-1. URL: https://www.sciencedirect.com/science/article/pii/S0022000076800451 (cit. on p. 169).

[Bod+16] Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michał Pilipczuk. "A $c^k n$ 5-Approximation Algorithm for Treewidth". In: *SIAM Journal on Computing* 45.2 (2016), pp. 317–378. DOI: 10.1137/130947374. eprint: https://doi.org/10.1137/130947374. URL: https://doi.org/10.1137/130947374 (cit. on p. 95).

[Bod98] Hans L Bodlaender. "A partial k-arboretum of graphs with bounded treewidth". In: *Theoretical computer science* 209.1-2 (1998), pp. 1–45 (cit. on p. 95).

[BR+18] Aniket Basu Roy, Sathish Govindarajan, Rajiv Raman, and Saurabh Ray. "Packing and Covering with Non-Piercing Regions". In: *Discrete & Computational Geometry* (2018) (cit. on pp. 8, 11, 26, 27, 141).

[Bra+11] Ulrik Brandes, Sabine Cornelsen, Barbara Pampel, and Arnaud Sallaberry. "Blocks of hypergraphs: applied to hypergraphs and outerplanarity". In: *Combinatorial Algorithms: 21st International Workshop, IWOCA 2010, London, UK, July 26-28, 2010, Revised Selected Papers 21*. Springer. 2011, pp. 201–211 (cit. on p. 8).

[Bra+12] Ulrik Brandes, Sabine Cornelsen, Barbara Pampel, and Arnaud Sallaberry. "Path-based supports for hypergraphs". In: *Journal of Discrete Algorithms* 14 (2012), pp. 248–261 (cit. on p. 8).

[BT11] Gerth Stølting Brodal and Konstantinos Tsakalidis. "Dynamic planar range maxima queries". In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2011, pp. 256–267 (cit. on pp. 122, 125).

[Buc+11] Kevin Buchin, Marc J van Kreveld, Henk Meijer, Bettina Speckmann, and KAB Verbeek. "On planar supports for hypergraphs". In: *Journal of Graph Algorithms and Applications* 15.4 (2011), pp. 533–549 (cit. on pp. 7, 8, 14, 39, 82, 83, 171, 172, 174).

[BYHR11] Reuven Bar-Yehuda, Danny Hermelin, and Dror Rawitz. "Minimum vertex cover in rectangle graphs". In: *Computational Geometry* 44.6-7 (2011), pp. 356–364 (cit. on p. 139).

[CAM15] Vincent Cohen-Addad and Claire Mathieu. "Effectiveness of Local Search for Geometric Optimization". In: *Proceedings of the Thirty-first International Symposium on Computational Geometry*. SoCG '15. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 329–343. ISBN: 978-3-939897-83-5 (cit. on p. 8).

[CF05] Raphaël Chand and Pascal Felber. "Semantic peer-to-peer overlays for publish/subscribe networks". In: *Euro-Par 2005 Parallel Processing: 11th International Euro-Par Conference, Lisbon, Portugal, August 30-September 2, 2005. Proceedings 11*. Springer. 2005, pp. 1194–1204 (cit. on p. 8).

[CG14a] Sergio Cabello and David Gajser. “Simple PTAS’s for families of graphs excluding a minor”. In: *CoRR* abs/1410.5778 (2014) (cit. on p. 11).

[CG14b] Timothy M. Chan and Elyot Grant. “Exact algorithms and APX-hardness results for geometric packing and covering problems”. In: *Comput. Geom. Theory Appl.* 47.2 (2014), pp. 112–124. DOI: 10.1016/j.comgeo.2012.04.001 (cit. on pp. 146–150).

[CH12] Timothy M. Chan and Sariel Har-Peled. “Approximation Algorithms for Maximum Independent Set of Pseudo-Disks”. In: *Discret. Comput. Geom.* 48.2 (2012), pp. 373–392. DOI: 10.1007/S00454-012-9417-5 (cit. on pp. 10, 11, 26, 27, 132–134, 137, 141).

[CH67] Gary Chartrand and Frank Harary. “Planar Permutation Graphs”. en. In: *Annales de l’institut Henri Poincaré. Section B. Calcul des probabilités et statistiques* 3.4 (1967), pp. 433–438. URL: https://www.numdam.org/item/AIHPB_1967__3_4_433_0/ (cit. on p. 66).

[Chv79] Vasek Chvatal. “A greedy heuristic for the set-covering problem”. In: *Mathematics of operations research* 4.3 (1979), pp. 233–235 (cit. on p. 131).

[CS89] Kenneth L Clarkson and Peter W Shor. “Applications of random sampling in computational geometry, II”. In: *Discrete & Computational Geometry* 4.5 (1989), pp. 387–421 (cit. on p. 135).

[Die05] Reinhard Diestel. “Graph theory 3rd ed”. In: *Graduate texts in mathematics* 173.33 (2005), p. 12 (cit. on p. 86).

[DN16] Zdeněk Dvořák and Sergey Norin. “Strongly Sublinear Separators and Polynomial Expansion”. In: *SIAM Journal on Discrete Mathematics* 30.2 (2016), pp. 1095–1101. DOI: 10.1137/15M1017569. eprint: https://doi.org/10.1137/15M1017569. URL: https://doi.org/10.1137/15M1017569 (cit. on p. 28).

[DRS02] Irit Dinur, Oded Regev, and Clifford Smyth. “The hardness of 3-Uniform hypergraph coloring”. In: Feb. 2002, pp. 33 –40. ISBN: 0-7695-1822-2. DOI: 10.1109/SFCS.2002.1181880 (cit. on p. 162).

[Du86] Ding-Zhu Du. “An optimization problem on graphs”. In: *Discrete applied mathematics* 14.1 (1986), pp. 101–104 (cit. on pp. 7, 8).

[Eve+03] Guy Even, Zvi Lotker, Dana Ron, and Shakhar Smorodinsky. “Conflict-Free Colorings of Simple Geometric Regions with Applications to Frequency Assignment in Cellular Networks”. In: *SIAM Journal on Computing* 33.1 (2003), pp. 94–136. DOI: 10.1137/S0097539702431840. eprint: https://doi.org/10.1137/S0097539702431840. URL: https://doi.org/10.1137/S0097539702431840 (cit. on p. 30).

[Fed87] Greg N Federickson. “Fast algorithms for shortest paths in planar graphs, with applications”. In: *SIAM Journal on computing* 16.6 (1987), pp. 1004–1022 (cit. on p. 9).

[Fei98] Uriel Feige. “A threshold of ln n for approximating set cover”. In: *Journal of the ACM (JACM)* 45.4 (1998), pp. 634–652 (cit. on p. 131).

[FG65] D. R. Fulkerson and O. A. Gross. "Incidence matrices and interval graphs." In: *Pacific Journal of Mathematics* 15.3 (1965), pp. 835 –855 (cit. on p. 169).

[FLR12] VZ Feinberg, AG Levin, and EB Rabinovich. *VLSI planarization: methods, models, implementation*. Vol. 399. Springer Science & Business Media, 2012 (cit. on p. 6).

[FP14] Jacob Fox and János Pach. "Applications of a New Separator Theorem for String Graphs". In: *Comb. Probab. Comput.* 23.1 (2014), pp. 66–74. DOI: 10.1017/S0963548313000412. URL: https://doi.org/10.1017/S0963548313000412 (cit. on p. 9).

[GHT84] John R Gilbert, Joan P Hutchinson, and Robert Endre Tarjan. "A separator theorem for graphs of bounded genus". In: *Journal of Algorithms* 5.3 (1984), pp. 391–407 (cit. on pp. 9, 11, 12, 26, 131).

[Gol80a] Martin Charles Golumbic. "CHAPTER 3 - Perfect Graphs". In: *Algorithmic Graph Theory and Perfect Graphs*. Ed. by Martin Charles Golumbic. Academic Press, 1980, pp. 51–80. ISBN: 978-0-12-289260-8. DOI: https://doi.org/10.1016/B978-0-12-289260-8.50010-8 (cit. on p. 124).

[Gol80b] Martin Charles Golumbic. "CHAPTER 5 - Comparability Graphs". In: *Algorithmic Graph Theory and Perfect Graphs*. Ed. by Martin Charles Golumbic. Academic Press, 1980, pp. 105–148. ISBN: 978-0-12-289260-8. DOI: https://doi.org/10.1016/B978-0-12-289260-8.50012-1 (cit. on pp. 124, 146).

[Gol+95] Paul W Goldberg, Martin C Golumbic, Haim Kaplan, and Ron Shamir. "Four strikes against physical mapping of DNA". In: *Journal of Computational Biology* 2.1 (1995), pp. 139–152 (cit. on p. 169).

[GOT17] Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, eds. *Handbook of Discrete and Computational Geometry*. 3rd ed. Boca Raton, FL: Chapman & Hall/CRC, 2017 (cit. on p. 14).

[GP10] Matt Gibson and Imran A. Pirwani. "Algorithms for Dominating Set in Disk Graphs: Breaking the logn Barrier". In: *Algorithms – ESA 2010*. Ed. by Mark de Berg and Ulrich Meyer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 243–254. ISBN: 978-3-642-15775-2 (cit. on pp. 10, 11).

[Hav+22] Frédéric Havet, Dorian Mazauric, Viet-Ha Nguyen, and Rémi Watrigant. "Overlaying a hypergraph with a graph with bounded maximum degree". In: *Discrete Applied Mathematics* 319 (2022), pp. 394–406 (cit. on p. 8).

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. "Expander graphs and their applications". In: *Bulletin of the American Mathematical Society* 43.4 (2006), pp. 439–561 (cit. on p. 9).

[HM85] Dorit S. Hochbaum and Wolfgang Maass. "Approximation schemes for covering and packing problems in image processing and VLSI". In: *J. ACM* 32.1 (Jan. 1985), 130–136. ISSN: 0004-5411. DOI: 10.1145/2455.214106. URL: https://doi.org/10.1145/2455.214106 (cit. on p. 10).

[Hos+12] Jun Hosoda, Juraj Hromkovič, Taisuke Izumi, Hirotaka Ono, Monika Steinová, and Koichi Wada. "On the approximability and hardness of minimum topic connected overlay and its special instances". In: *Theoretical Computer Science* 429 (2012), pp. 144–154 (cit. on p. 8).

[H99] J Håstad. "Clique is hard to approximate within $n^{1-\varepsilon}$". In: *Acta Mathematica* 182.1 (1999), pp. 105–142 (cit. on p. 131).

[Hur+18] Ferran Hurtado, Matias Korman, Marc van Kreveld, Maarten Löffler, Vera Sacristán, Akiyoshi Shioura, Rodrigo I Silveira, Bettina Speckmann, and Takeshi Tokuyama. "Colored spanning graphs for set visualization". In: *Computational Geometry* 68 (2018), pp. 262–276 (cit. on p. 8).

[JP87] David S Johnson and Henry O Pollak. "Hypergraph planarity and the complexity of drawing Venn diagrams". In: *Journal of graph theory* 11.3 (1987), pp. 309–325 (cit. on pp. xi, 6–8, 14, 39, 171, 174).

[Kes20] Balázs Keszegh. "Coloring Intersection Hypergraphs of Pseudo-Disks". In: *Discret. Comput. Geom.* 64.3 (2020), pp. 942–964. DOI: 10.1007/s00454-019-00142-6. URL: https://doi.org/10.1007/s00454-019-00142-6 (cit. on pp. 29, 30, 151).

[KKS09] Michael Kaufmann, Marc van Kreveld, and Bettina Speckmann. "Subdivision drawings of hypergraphs". In: *Graph Drawing: 16th International Symposium, GD 2008, Heraklion, Crete, Greece, September 21-24, 2008. Revised Papers 16*. Springer. 2009, pp. 396–407 (cit. on pp. 7, 8).

[KP18] Balázs Keszegh and Dömötör Pálvölgyi. "Aligned plane drawings of the generalized Delaunay-graphs for pseudo-disks". In: *J. Comput. Geom.* 11 (2018), pp. 354–370. URL: https://api.semanticscholar.org/CorpusID:263873367 (cit. on p. 14).

[KP19] Balázs Keszegh and Dömötör Pálvölgyi. "An abstract approach to polychromatic coloring: shallow hitting sets in ABA-free hypergraphs and pseudohalfplanes". In: *J. Comput. Geom.* 10.1 (2019), pp. 1–26. DOI: 10.20382/jocg.v10i1a1. URL: https://doi.org/10.20382/jocg.v10i1a1 (cit. on p. 159).

[Kro+14] Erik Krohn, Matt Gibson, Gaurav Kanade, and Kasturi Varadarajan. "Guarding terrains via local search". In: *Journal of Computational Geometry* 5.1 (2014), pp. 168–178 (cit. on p. 8).

[KS03] Ephraim Korach and Michal Stern. "The clustering matroid and the optimal clustering tree". In: *Mathematical Programming* 98 (2003), pp. 385–414 (cit. on p. 8).

[KS18] Chaya Keller and Shakhar Smorodinsky. "Conflict-Free Coloring of Intersection Graphs of Geometric Objects". In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*. 2018, pp. 2397–2411 (cit. on pp. 29, 30, 151).

[LN23] Seunghun Lee and Eran Nevo. *On colorings of hypergraphs embeddable in* $\mathbb{R}^d$. 2023. arXiv: 2307.14195 [math.CO]. URL: https://arxiv.org/abs/2307.14195 (cit. on p. 159).

[Lov75] Laszlo Lovász. "On the ratio of optimal integral and fractional covers". In: *Discrete Math* 13 (1975), p. 383390 (cit. on p. 131).

[LT79] Richard J. Lipton and Robert E. Tarjan. "A separator theorem for planar graphs". In: *SIAM Journal on Applied Mathematics* 36.2 (1979), pp. 177–189 (cit. on p. 9).

[LT80] Richard J Lipton and Robert Endre Tarjan. "Applications of a planar separator theorem". In: *SIAM journal on computing* 9.3 (1980), pp. 615–627 (cit. on p. 9).

[Mat02] Jiří Matoušek. *Lectures on Discrete Geometry*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002. ISBN: 0387953744 (cit. on pp. 12, 78).

[MR10] Nabil H Mustafa and Saurabh Ray. "Improved results on geometric hitting set problems". In: *Discrete & Computational Geometry* 44.4 (2010), pp. 883–895 (cit. on pp. 8, 10, 11, 26, 27, 132, 134, 137, 141, 145, 146).

[MRR15] Nabil H. Mustafa, Rajiv Raman, and Saurabh Ray. "Quasi-Polynomial Time Approximation Scheme for Weighted Geometric Set Cover on Pseudodisks and Halfspaces". In: *SIAM Journal on Computing* 44.6 (2015), pp. 1650–1669 (cit. on p. 131).

[MT01] Bojan Mohar and Carsten Thomassen. "Graphs on Surfaces". In: *Johns Hopkins series in the mathematical sciences*. 2001. URL: https://api.semanticscholar.org/CorpusID:27349785 (cit. on pp. 22, 29, 39, 140, 151).

[NM12] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity: graphs, structures, and algorithms*. Springer Publishing Company, Incorporated, 2012 (cit. on p. 9).

[NTJ75] George L Nemhauser and Leslie E Trotter Jr. "Vertex packings: structural properties and algorithms". In: *Mathematical Programming* 8.1 (1975), pp. 232–248 (cit. on p. 139).

[Opa79] Jaroslav Opatrny. "Total Ordering Problem". In: *SIAM Journal on Computing* 8.1 (1979), pp. 111–114. DOI: 10.1137/0208008 (cit. on p. 170).

[OR11] Melih Onus and Andréa W Richa. "Minimum maximum-degree publish–subscribe overlay network design". In: *IEEE/ACM Transactions on Networking* 19.5 (2011), pp. 1331–1343 (cit. on p. 8).

[PR08] Evangelia Pyrga and Saurabh Ray. "New Existence Proofs for ϵ-nets". In: *Proceedings of the Twenty-fourth Annual Symposium on Computational Geometry*. SoCG '08. New York, NY, USA: ACM, 2008, pp. 199–207. ISBN: 978-1-60558-071-5 (cit. on p. 145).

[Pá24] Dömötör Pálvölgyi. "Note on Polychromatic Coloring of Hereditary Hypergraph Families". In: *Graphs and Combinatorics* 40.6 (Nov. 22, 2024), p. 131. ISSN: 1435-5914. DOI: 10.1007/s00373-024-02836-y. URL: https://doi.org/10.1007/s00373-024-02836-y (cit. on p. 12).

[RR20] Rajiv Raman and Saurabh Ray. "Constructing planar support for non-piercing regions". In: *Discrete & Computational Geometry* 64.3 (2020), pp. 1098–1122 (cit. on pp. 8, 11, 13, 15, 17, 21, 23, 25–30, 37, 42, 63, 78, 79, 102, 103, 125, 134, 141, 146, 151, 154, 171, 175).

[RR22] Rajiv Raman and Saurabh Ray. "On the Geometric Set Multicover Problem". In: *Discret. Comput. Geom.* 68.2 (2022), pp. 566–591. DOI: 10.1007/s00454-022-00402-y (cit. on p. 135).

[RS04] Neil Robertson and P.D. Seymour. "Graph Minors. XX. Wagner's conjecture". In: *Journal of Combinatorial Theory, Series B* 92.2 (2004). Special Issue Dedicated to Professor W.T. Tutte, pp. 325–357. ISSN: 0095-8956. DOI: https://doi.org/10.1016/j.jctb.2004.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0095895604000784 (cit. on p. 4).

[RS86] Neil Robertson and P.D Seymour. "Graph minors. II. Algorithmic aspects of tree-width". In: *Journal of Algorithms* 7.3 (1986), pp. 309–322. ISSN: 0196-6774. DOI: https://doi.org/10.1016/0196-6774(86)90023-4. URL: https://www.sciencedirect.com/science/article/pii/0196677486900234 (cit. on pp. 24, 85).

[RY68] Gerhard Ringel and J. W. T. Youngs. "Solution of the Heawood map-coloring problem". In: *Proceedings of the National Academy of Sciences of the USA* 60 (1968), pp. 438–445 (cit. on pp. 29, 151).

[Sau72] N Sauer. "On the density of families of sets". In: *Journal of Combinatorial Theory, Series A* 13.1 (1972), pp. 145–147. ISSN: 0097-3165. DOI: https://doi.org/10.1016/0097-3165(72)90019-2. URL: https://www.sciencedirect.com/science/article/pii/0097316572900192 (cit. on p. 78).

[Sha03] Micha Sharir. "The Clarkson–Shor technique revisited and extended". In: *Combinatorics, Probability and Computing* 12.2 (2003), pp. 191–201 (cit. on p. 135).

[SY12] Shakhar Smorodinsky and Yelena Yuditsky. "Polychromatic coloring for half-planes". English. In: *J. Comb. Theory, Ser. A* 119.1 (2012), pp. 146–154. ISSN: 0097-3165. DOI: 10.1016/j.jcta.2011.07.001 (cit. on p. 159).

[TY84] Robert E Tarjan and Mihalis Yannakakis. "Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs". In: *SIAM Journal on computing* 13.3 (1984), pp. 566–579 (cit. on pp. 7, 8).

[VF84] AA Voloshina and VZ Feinberg. "PLANARITY OF HYPERGRAPHS". In: *DOKLADY AKADEMII NAUK BELARUSI*. Vol. 28. ACADEMII NAUK BELARUSI F SCORINA PR 66 ROOM 403 MINSK BYELARUS 220072. 1984, pp. 309–311 (cit. on pp. xi, 4, 6).

[VV04] Anne Verroust and Marie-luce Viaud. "Results on hypergraph planarity". In: (Jan. 2004) (cit. on p. 6).

[Wei16] Eric W Weisstein. "Torus grid graph". In: *https://mathworld. wolfram. com/* (2016) (cit. on p. 41).

[WZ90] Sue Whitesides and Rongyao Zhao. *K-admissible collections of Jordan curves and offsets of circular arc figures*. McGill University, School of Computer Science, 1990 (cit. on p. 10).

[Zyk74] Alexander Aleksandrovich Zykov. "Hypergraphs". In: *Russian Mathematical Surveys* 29.6 (1974), p. 89 (cit. on pp. xi, 6).