Karlsruhe Institute of Technology

On Hyperbolic Unit Disk Graphs

Master Thesis of

Emil Dohse

At the Department of Informatics
Institute of Theoretical Informatics

Reviewers and Advisors:

Thomas Bläsius
Torsten Ueckerdt

Time Period: 1st September 2021 – 1st March 2022

Statement of Authorship

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, February 14, 2022

Abstract

The study of models for large real-world networks is an integral part of network science and of ever-increasing practical relevance in the context of networks like the internet. For that, a good model should be similar to real-world networks but at the same time allow for mathematical analysis. Papadopoulos, Krioukov, Boguná and Vahdat [PKBV10] introduced such a model – *hyperbolic random graphs*. The graphs produced by this model are *strongly hyperbolic unit disk graphs (SHDGs)*, that is, they are intersection graphs of disks with an arbitrary radius R in hyperbolic space, where the disks are placed randomly in a ground space limited by a disk of radius R. It has been shown, that a graph generated by this random graph model resembles many real-world networks in regard to their properties with high probability. Very little work exists for the graph class itself though and it is not always clear if the results are due to properties of all SHDGs or due to the distribution the disks are sampled from.

In this thesis, we study *hyperbolic unit disk graphs (HDGs)* with unlimited ground space and the subclass SHDGs in a non-probabilistic setting. This is done to get worst-case guarantees for algorithms and graph properties, and to better understand the underlying structure induced by hyperbolic geometry. We relate the graph class of HDGs to similar graph classes like SHDGs, Euclidean unit disk graphs (EDGs), circular arc graphs and chordal graphs. For HDGs we show that recognition is NP-hard and a member of $\exists\mathbb{R}$. We further contribute an algorithm to find maximum cliques in SHDGs without knowing an embedding of the disks in the plane, with run time $\mathcal{O}(\sum_{v \in V} |N(v)|^3)$ where $|N(v)|$ is the degree of node v. For HDGs we show that maximum cliques can be found efficiently with an algorithm originally designed for EDGs [RS03] but also that there is a gap in the proof of this algorithm that we were not able to fix. Lastly, we discuss grids in SHDGs and contribute some structural results on separators and treewidth. These lead to an efficient algorithm that generates a tree decomposition for an SHDG of width $4\Delta \log(n)$, where Δ is the maximum degree.

Deutsche Zusammenfassung

Die mathematische Untersuchung von Modellen für große Echtwelt-Netzwerke ist ein integraler Bestandteil der *Network Science* und wird durch die gesellschaftliche Bedeutung von Netzwerken wie dem Internet in der Praxis immer wichtiger. Ein gutes Modell sollte daher realen Netzen ähnlich sein, aber gleichzeitig eine mathematische Analyse ermöglichen. Die von Papadopoulos, Krioukov, Boguná and Vahdat [PKBV10] eingeführten *hyperbolic random graphs* sind ein solches Modell. Die Graphen, die von diesem Modell erzeugt werden, sind *strongly hyperbolic unit disk graphs* (SHDGs), d.h., sie sind Schnittgraphen von Kreisscheiben mit beliebigem Radius R in der hyperbolischen Ebene, wobei die Mittelpunkte der Kreisscheiben zufällig in einem durch eine Kreisscheibe mit Radius R begrenzten Grundraum verteilt sind. Ein mit diesem Zufallsgraphenmodell erzeugter Graph hat mit hoher Wahrscheinlichkeit viele der Eigenschaften, die auch reale Netzwerke aufweisen. Es gibt jedoch nur wenige Arbeiten zu der Graphenklasse selbst und bei den bereits gezeigten Aussagen ist nicht immer klar, ob sie auf Eigenschaften aller SHDGs zurückzuführen sind oder die Verteilung, mit der die Graphen erzeugt wurden, ausschlaggebend war.

In dieser Masterarbeit untersuchen wir *hyperbolic unit disk graphs (HDGs)* mit unbeschränktem Grundraum und die Unterklasse SHDGs in einem nicht-probabilistischen Rahmen. Dadurch bekommen wir Worst-Case-Garantien für Algorithmen und Grapheigenschaften und somit ein besseres Verständnis für die zugrundeliegende Struktur, die durch die hyperbolische Geometrie induziert wird. Wir vergleichen die Graphenklasse von HDGs mit ähnlichen Graphenklassen wie SHDGs, *Euclidean unit disk graphs (EDGs)*, Kreisbogengraphen und Chordalen Graphen. Für HDGs zeigen wir, dass die Erkennung NP-schwer und ein Mitglied von $\exists\mathbb{R}$ ist. Wir stellen außerdem einen Algorithmus mit einer Laufzeit von $\mathcal{O}(\sum_{v\in V} |N(v)|^3)$ vor, um maximale Cliquen in SHDGs zu finden, ohne eine Einbettung der Kreisscheiben in der Ebene zu kennen. Dabei ist $|N(v)|$ der Grad des Knotens v. Für HDGs zeigen wir, dass maximale Cliquen effizient mit einem Algorithmus gefunden werden können, der ursprünglich für EDGs [RS03] entworfen wurde. Wir haben jedoch eine Lücke im Beweis dieses Algorithmus gefunden, die wir nicht schließen konnten. Schließlich diskutieren wir Gitter in SHDGs und stellen einige strukturelle Ergebnisse zur Baumweite sowie einen effizienten Algorithmus vor, der eine Baumzerlegung der Weite $4\Delta \log(n)$ für einen SHDG berechnet, wobei Δ der maximale Grad eines Knotens ist.

Contents

1. Introduction

Large networks have become more and more important. Be it social networks, the interaction networks of proteins or the internet, as a network crucially important for today's society. Solving routing problems on the internet or predicting the interactions of proteins just based on their network are problems of practical significance. To solve these problems, one wants to exploit structural properties of the networks at hand. Network science has come to help – understanding the real word networks, finding theoretical models similar to these real networks and using the well-defined structure of the models to invent and improve algorithms that then can also be applied to real-world networks. The above-mentioned problems about protein interactions and routing on the internet were solved solely based on the properties of their respective networks [Kle07, KLS+19]. Ducan Watts has gone so far to say: "Networks are important because if we don't understand networks, we can't understand how markets function, organizations solve problems, or how societies change." [BRMW13][1].

This thesis is on better understanding one of the theoretical models facilitating properties of real networks – *hyperbolic unit disk graphs*. Hyperbolic unit disk graphs (HDGs) are the intersection graphs of disks with an arbitrary radius R in hyperbolic space. So far, mostly a subclass of HDGs has been studied, in which the midpoints of the disks are randomly placed in a disk with radius R, the *ground space*. These so-called *hyperbolic random graphs*, introduced by Papadopoulos *et al.* [PKBV10], have been shown to fulfill many of the characteristic properties of real-world networks, such as a power-law degree distribution [PKBV10], constant clustering coefficient [GPP12], small diameter [FK18] and very small average distances [ABF15]. The properties mentioned above are shown to hold with high probability when the graph is generated by randomly placing the midpoints of the disks in the ground space. This is a natural thing to do for random generated graphs and often yields explanations for run times in practical applications. It has the downside though, that algorithms possibly run way slower on some instances and that the properties of the graphs depend highly on the chosen distribution from which the positions of the midpoints are sampled. We consider the graph class of HDGs in a non-probabilistic setting to get worst-case guarantees for algorithms and graph properties and to better understand the underlying structure induced by hyperbolic geometry.

In a non-probabilistic setting, HDGs have only been considered by Kisfaludi-Bak [KB20] (they consider noisy unit ball graphs in arbitrary dimension, but the results hold in the

[1] Originally said in a televised documentary on *Australian Broadcasting Corporation* (ABC), Australia, in 2008 and referred to by Brandes *et al.* [BRMW13].

worst-case). They examine the structure of HDGs parameterized by the radius though, which is a severe restriction since unlike in Euclidean unit disk graphs, the radius of the disks plays an important role in what graphs can be embedded. Additionally, Bläsius *et al.* [BFKS21] made the argument, that for a graph with constant average degree and fixed radius, a growing number of nodes forces the graph to be embedded in growing regions of hyperbolic space. This in turn leads to a high diameter of the graph and a more grid-like structure which is usually observed in Euclidean unit disk graphs.

As a counterpart to these grid-like structures, Bläsius *et al.* [BFKS21] further motivates the study of the subclass of HDGs limited to a ground space of radius R in a non-probabilistic setting. By limiting the ground space, the graphs have a hierarchical structure – a property seen in many real-world networks like the internet. They name this graph class *strongly hyperbolic unit disk graphs* (SHDGs) what we adopt in this thesis.

The contribution of this thesis is to make the first steps in relating the class of HDGs and the subclass of SHDGs to similar graph classes. This is done in Chapter 3. The natural candidate to compare with are Euclidean unit disk graphs (EDGs) [CCJ90], a well-known graph class for which the definition *only* differs from the definition of HDGs in the geometry of the embedding. We show that EDGs are a sub-class of HDGs and that SHDGs and EDGs are incomparable.

Another candidate to compare with are chordal graphs [BP93]. They are similar in having a perfect elimination ordering compared to HDGs having an elimination ordering in which the neighborhood of the endpoints of an edge forms two cliques. We show that chordal graphs and HDGs are incomparable. We also relate HDGs and SHDGs to circular arc graphs [Tuc70]. While circular arc graphs are intersection graphs of circular arcs at the boundary of a disk, SHDGs are intersection graphs of disks within a disk, that can be placed to all have midpoints with the same radius. Using this, we show that unit circular arc graphs are SHDGs which also shows that unit interval graphs and proper interval graphs are SHDGs. Furthermore, we show that SHDGs are no circular arc graphs in general.

In Chapter 4 we discuss the problem of recognizing HDGs. We use the similarities between EDGs and HDGs to show that recognition of HDGs is NP-hard. Recognition of EDGs was shown to be $\exists\mathbb{R}$-complete [KM12]. We provide the first step to possibly show this for recognizing HDGs as well by proving it is a member of $\exists\mathbb{R}$.

In Chapter 5 we then proceed to discuss maximum cliques in unit disk graphs. For both Euclidean [CCJ90] and hyperbolic [BFK18] unit disk graphs there is a polynomial-time algorithm for finding all maximum cliques when an embedding is given, i.e., the coordinates of the centers of the disk in the plane. If there is no embedding given, Raghavan *et al.* [RS03] gave proof of there being a polynomial-time algorithm as well. We found a gap in one of their arguments though that we were not able to fix. This makes it unclear whether a polynomial-time algorithm to find all maximum cliques without an embedding exists. We further show that if their algorithm is proven to be correct, it works for HDGs as well, not only for EDGs. Independent of their result, we contribute an algorithm for finding all maximum cliques in SHDGs without knowing an embedding with run time $\mathcal{O}(\sum_{v \in V} |N(v)|^3)$ where $|N(v)|$ is the degree of node v.

In Chapter 6 we move on to examine separators, grids and treewidth in SHDGs. The treewidth of a graph is an important parameter. Informally speaking it quantifies how similar to a tree a graph is. When the treewidth is small, many NP-hard graph problems can be solved efficiently. This often works by first finding a so-called tree decomposition for the graph and then using dynamic programming on this tree decomposition. It is important for the efficiency that the tree decomposition has a small *width*, optimally the treewidth. Tree decompositions are closely related to balanced separators, i.e., node sets

that, when removed, split the remaining graph into two sets of similar size. We present an algorithm that efficiently finds balanced separators of size 4Δ or less, where Δ is the maximum degree. The algorithm can then be applied to get a tree decomposition of similar width. We further discuss grids in SHDGs. We conjecture that while keeping the clique number constant, an SHDG can have a grid minor of any size. This comes at the cost of also having exponentially many nodes though.

2. Preliminaries

In this thesis $G = (V, E)$ is an undirected, simple *graph*, where the set V are the *nodes* (or sometimes *vertices*) and the set $E \subseteq \{\{u, v\} \mid u \neq v, u, v \in V\}$ are the *edges*. For an edge $\{u, v\}$, the nodes u and v are called the *endpoints*. For a graph, we usually denote the number of nodes $|V|$ by n and the number of edges $|E|$ by m. Let u, v be nodes, node u is *adjacent* to node v, if $\{u, v\} \in E$. The *neighborhood* $N(v)$ of a node $v \in V$ are the nodes adjacent to v and the node itself, i.e., $N(v) = \{u \mid \{u, v\} \in E\} \cup \{v\}$. Let $e = \{u, v\} \in E$ be an edge, we call the set $N(u) \cap N(v)$ the *(shared) neighborhood* of e. The *degree* of a node v is the cardinality of its neighborhood, i.e., $|N(v)|$.

We call a graph $G' = (V', E')$ a *subgraph* of G, if $V' \subseteq V$ and $E' \subseteq E$. We call G' an *induced subgraph*, if it is a subgraph and all edges present in G between nodes in G' are also present in G', i.e., $\{u, v\} \in E \wedge u, v \in V' \Rightarrow \{u, v\} \in E'$. For a subset of the nodes $V' \subseteq V$ we denote by $G[V']$ the induced subgraph of G with node set V'. For a subset $U \subseteq V$ we denote by $G - U$ the induced subgraph of G with vertex set $V \setminus U$. We say a graph G' is a *minor* of G, if it can be formed from G by a series of edge deletions, node deletions (where the node is not part of any edge anymore) and *edge contractions*, i.e., an edge $\{u, v\}$ is removed and its endpoints are merged, yielding a single node w that replaces u and v as endpoints in every edge they were a part of. The *complement graph* $\overline{G} = (V, \overline{E})$ of G is the graph that contains all the edges not present in G but none of the edges present in G, i.e., $e \in E \Leftrightarrow e \notin \overline{E}$.

A *path* is a finite sequence of edges $e_1, ..., e_t$, such that e_i and e_{i+1} share one node for $1 \leq i < t$. Furthermore, each node is an endpoint of at most two edges on the path. We say a node is *on the path* if it is an endpoint of any edge of the path. The *length* of a path is the number of edges in the sequence. The *distance* between two nodes u, v in a graph G is the length of the shortest path in G, such that u only is an endpoint of the first edge and v only is an endpoint of the last edge. We say two nodes are *connected* in G if there is a path with both nodes on it. A *connected component* of G is an induced subgraph of G with nodes $V' \subseteq V$, such that all nodes in V' are pairwise connected and there is no node in $V \setminus V'$ that is connected to a node in V'. A *cycle* is a set of nodes $\{v_0, ..., v_i\}$ such that nodes v_k and $v_{k+1 \bmod i}$ for $0 \leq k \leq i$, are adjacent. If G consists of a single cycle, we say G is a *cycle graph*. The cycle graph with n vertices is denoted by C_n. We say a graph has an *induced cycle* if it contains a cycle graph as an induced subgraph. A *tree* is a graph that does not contain a cycle graph as a subgraph. For a fixed node r of a tree T, we call (T, r) a *rooted tree* with root r. The *height* of a rooted tree is the maximum distance from a node of the tree to the root. For a rooted tree (T, r) and nodes u, v of T, we say u is an *ancestor* of v if the path form r to v in T has u on it. In that case we calls v a *descendant* of u in T. If a descendant of u is adjacent to u, we call it a *child*, if an ancestor of v is adjacent to v we call it its *parent*. Nodes of T that do not have descendants are called *leaves*. A *clique* is a set of nodes $K \subseteq V$, such that the nodes are pairwise adjacent, i.e., $u, v \in K, u \neq v \Rightarrow \{u, v\} \in E$. If all nodes of G form a clique, we say G is the *complete graph*. We denote the complete graph with n nodes by K_n. An *independent set* is a set of nodes, such that none of the nodes are adjacent. We say a node v is an *independent node with respect to a set of nodes $N \subseteq V$* if v is not adjacent to any node $n \in N$. If it is clear from the context what the set N is, we just say v is an *independent node*. A *vertex cover $W \subseteq V$* is a set of nodes, such that every edge has an endpoint in W. A *matching $M \subseteq E$* is a set of edges, such that no two edges in M share a node. A *star graph S_i* is the graph $i + 1$ nodes where one node is adjacent to i independent nodes. A graph is *bipartite* if its nodes can be partitioned into two independent sets. We call a graph G a *$t \times t$ grid*, if there is a bijection $b : V \to \{1, ..., t\} \times \{1, ..., t\}$ between the set of nodes and an index set such that two nodes u, v with $b(u) = (x, y)$ and $b(v) = (x', y')$ are adjacent, if and only if $|x - x'| + |y - y'| = 1$. We say a graph contains a grid of size s when it contains a $s \times s$ grid as a minor.

An *interval graph* is a graph $G = (V, E)$ for which a family of intervals $\mathcal{I} := \{I_v | v \in V\}$ exists such that for nodes $u, v \in V$ the intervals I_u, I_v intersect if and only if the nodes are adjacent, i.e., $\{u, v\} \in E$. We call $\mathcal{I}$ the *interval model* of G. A *unit interval graph G* is an interval graph for which there is an interval model of G in which all intervals have the same length. A *proper interval graph G* is an interval graph for which there is an interval model of G in which no interval fully contains another interval. A *circular arc graph* is a graph $G = (V, E)$ for which a family of circular arcs $\mathcal{A} := \{A_v | v \in V\}$ exists such that for nodes $u, v \in V$ the arcs A_u, A_v intersect if and only if the nodes are adjacent, i.e., $\{u, v\} \in E$. We call $\mathcal{A}$ the *arc model* of G. A *unit circular arc graph G* is an circular arc graph for which there is an arc model of G in which all arcs have the same length.

The hyperbolic plane. To be able to talk about points in the hyperbolic plane $\mathbb{H}^2$, let o be a hyperbolic point that we call the *origin*. We further choose a ray starting at o for which we say it has angle 0 and all angles around o are relative to this ray. Now we identify every point in the hyperbolic plane by its distance from o which we call *radius* and its angle around o, relative to the ray we chose, i.e., we get polar coordinates for the hyperbolic plane with origin o. For two points $p = (r_1, \theta_1)$ and $q = (r_2, \theta_2)$ with radii r_1 and r_2, and angles θ_1 and θ_2, the hyperbolic distance function d_h is given by

$$d_h((r_1, \theta_1), (r_2, \theta_2)) = \operatorname{arcosh}(\cosh(r_1)\cosh(r_2) - \sinh(r_1)\sinh(r_2)\cos(\Delta_\theta(\theta_1, \theta_2))),$$

where Δ_θ is the *angular distance*, i.e., $\Delta_\theta(\theta_1, \theta_2) = \min(|\theta_1 - \theta_2|, ||\theta_1 - \theta_2| - 2\pi|)$.

Informally speaking, hyperbolic geometry mainly differs from Euclidean geometry in that it has much more space. More formally, while the area of a Euclidean circle grows quadratically in the radius, in a hyperbolic cycle the area grows exponentially in the radius. Along with other reasons, this makes it hard to visualize hyperbolic space in Euclidean space. To represent the different properties of hyperbolic geometry, there are multiple equivalent models, each with an emphasis on certain aspects but none able to visualize all properties at once. For a brief overview of hyperbolic geometry, we refer to the article [KPK$^+$10], which also gives further references. In this thesis, we mainly consider the hyperbolic plane in two of the representations. A third is introduced in Section 4.2 on $\exists\mathbb{R}$.

The representation we use the most is the *native representation*. It is illustrated in Fig. 2.1 on the left. The hyperbolic coordinates are interpreted as Euclidean polar coordinates. Thus, the hyperbolic radius of a point is its Euclidean radius in the representation as well.

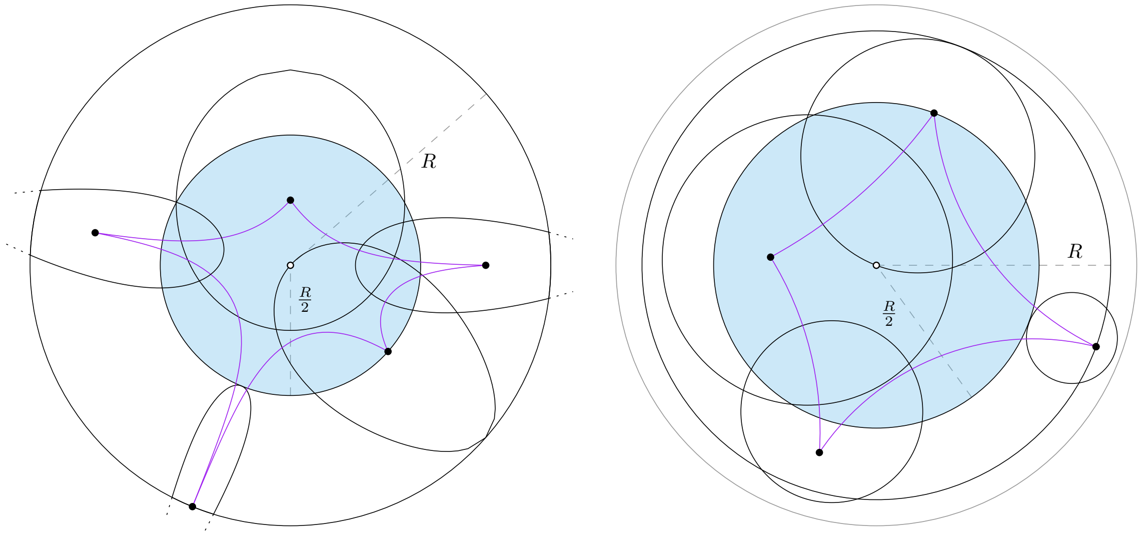

Figure 2.1.: Native representation on the left, Poincaré disk representation on the right. Disks with radius $\frac{R}{2}$ in ground space of radius R, displayed together with their midpoints in black and line segments between the midpoints in purple.

Circles have a tear-like form, dropping towards the origin. Compared to Euclidean lines, hyperbolic lines are distorted towards the origin.

The other representation we use is the *Poincaré disk model.* It is illustrated in Fig. 2.1 on the right. The hyperbolic plane is embedded into a unit disk in Euclidean space. The boundary of the disk does not represent any hyperbolic points but can be considered to be hyperbolic points with radius ∞. In this model, straight lines are circular arcs and circles stay circles but their radius depends on the distance of their centerpoint from the origin, and they get very small quite quickly.

Unit disk graphs. Similar to d_h above, we denote the Euclidean distance function by d_e. When it is clear from the context whether we use the Euclidean or hyperbolic distance, we simply denote the distance function by d. For a point p in the Euclidean or hyperbolic plane we define the *disk* $B_p(r)$ *with radius* r and *mid- or centerpoint* p to be the set of points with distance smaller than or equal to r of p, i.e., $B_p(r) = \{q \in \mathcal{X} \mid d(p,q) \leq r\}$, where $\mathcal{X}$ is either $\mathbb{R}^2$ or $\mathbb{H}^2$, depending on the origin of p.

An *embedding* of a graph $G = (V, E)$ is a function $f_e : V \to \mathbb{R}^2$ or $f_h : V \to \mathbb{H}^2$ mapping the vertices of G to points in the Euclidean or hyperbolic plane. Let u, v be nodes, for fixed embeddings f_e, f_h in the Euclidean and hyperbolic plane, we say the nodes u and v have distance $d_e(f_e(u), f_e(v))$ or $d_h(f_h(u), f_h(v))$, respectively. A graph G is a *Euclidean unit disk graph (EDG)* if it admits an embedding e in the Euclidean plane, such that two nodes u, v are adjacent if and only if u and v have distance smaller than or equal to 1, i.e., $\{u, v\} \in E \Leftrightarrow d_e(e(u), e(v)) \leq 1$. Similarly, we say a graph G is a *hyperbolic unit disk graph (HDG)* if it admits an embedding h in the hyperbolic plane, such that two nodes u, v are adjacent if and only if u and v have distance smaller than or equal to an arbitrary but fixed $R \in \mathbb{R}$, i.e., $\{u, v\} \in E \Leftrightarrow d_h(h(u), h(v)) \leq R$. We call R the *radius* or *threshold distance* of the embedding. Observe, while for EDGs the chosen distance does not make a difference for what graphs are EDGs, in the hyperbolic case it makes a difference as we will see in this thesis. As a subclass of HDGs, we say a graph G is a *strongly hyperbolic unit disk graph (SHDG)* if it is an HDG with an embedding h with threshold distance R, such that all nodes are embedded in the disk with radius R around the hyperbolic origin o, i.e., in $B_o(R)$. Given the embedding e of a unit disk graph $G = (V, E)$, the *inner neighborhood* of a node $v \in V$ are the nodes $u \in N(v)$ in the neighborhood of v that have a radius smaller than or equal to the radius of v in the given embedding.

When talking about nodes of a unit disk graph with a given embedding, we sometimes abuse notation and treat the nodes as the points they are embedded at, e.g. for an SHDG G with given embedding e and nodes u, v, we say that the nodes u and v have distance r if $d(e(u), e(v)) = r$ or we say H is the convex hull of u and v if H is the convex hull of $e(u), e(v)$.

Representations of unit disk graphs. We use two different visualizations of the embedding of a unit disk graph. The intersection representation and the inclusion representation.

For an HDG G with an embedding e that has radius R, we know that all adjacent nodes are embedded with distance at most R. Let u, v be adjacent nodes of G, then the disks with radius $\frac{R}{2}$ around $e(u)$ and $e(v)$, i.e., $B_{e(u)}(\frac{R}{2})$ and $B_{e(v)}(\frac{R}{2})$, intersect. Therefore, we call this representation with disks $B_{e(v)}(\frac{R}{2})$ for each node $v \in V$ *intersection representation.* In the intersection representation two nodes are adjacent if and only if their disks intersect. This is also why unit disk graphs are called unit disk graphs, they are intersection graphs of disks of unit radius.

The *inclusion representation* uses disks of radius R for each node instead. Thus, for a node v of G, the disk $B_{e(v)}(R)$ contains each of the points $e(u)$ for $u \in N(v)$, i.e., it *includes* the set $\{e(u) \mid u \in N(v)\}$.

To visualize either of the representations, the boundaries of the disks are depicted.

Geometry. We need some concepts regardless of the geometry being the hyperbolic or Euclidean geometry. For each of the following definitions, the points are either from the Euclidean or the hyperbolic plane. When talking about the shortest way to get from a point a to a point b, we say the *geodesic* between a and b for the curve along the line that a and b are on. A (usually infinite) set P of points is said to be *convex* if the geodesic between any two points of P lies in P. Let K be a finite or infinite set of points. A convex set P is called the convex hull of K, if P contains K, i.e., $K \subseteq P$, and there is no convex set of points $P' \subset P$ that still contains K.

For three points $p = (r_p, \alpha)$, $q = (r_q, \beta)$ and $x = (r_x, \gamma)$ with radii r_p, r_q, r_x and angles α, β, γ, we say x lies *in between* p and q if $\Delta_\theta(\alpha, \beta) = \Delta_\theta(\alpha, \gamma) + \Delta_\theta(\gamma, \beta)$ where Δ_θ is the angular distance.

Lemma 2.1. *(Lemma 5.4 in [Kro16]). Let $G = (V, E)$ be an SHDG. Let $u, v, w \in V$ be vertices with radii r_u, r_v and r_w, respectively, such that v lies in between u and w, and let $\{u, w\} \in E$. If $r_v \leq r_u$, then v is adjacent to w.*

Lemma 2.2. *Let $G = (V, E)$ be an SHDG and let $w \in V$ be a node. The inner neighborhood of w consists of two cliques.*

Proof. Let the angle of w be π without loss of generality and the radius be r_w. We show that the inner neighbors with angles larger than π form a clique, the proof for the inner neighbors with angles less than or equal to π goes analogously. Let u and v be nodes in the inner neighborhood of w with angle larger than π. Let their radii be r_u and r_v, respectively. Without loss of generality, v lies in between u and w. Since v is an inner neighbor of w, it has a radius smaller than or equal to the radius of w, i.e., $r_v \leq r_w$, and is adjacent to u according to Lemma 2.1. Thus, all inner neighbors of w with radius larger than π are pairwise adjacent, i.e., they form a clique. □

Lemma 2.3. *Let $p = (a, 0)$ and $q = (b, \alpha)$ with $\alpha \in (0, \pi]$ be points in the hyperbolic plane. Increasing the radius of both points by r at the same time increases the distance of the points.*

Proof. The hyperbolic distance of the points is given by $d(p, q) = \operatorname{arcosh}(\cosh(a)\cosh(b) - \sinh(a)\sinh(b)\cos(\alpha))$. We want to know how the distance changes when we incease the radius. To that end, we omit *arcosh*, since it is a strictly monotone function and look at $\cosh(x + a)\cosh(x + b) - \sinh(x + a)\sinh(x + b)\cos(\alpha)$, where x is the change of radius. We derive:

$$\begin{aligned}
&\frac{d}{dx}\cosh(x + a)\cosh(x + b) - \sinh(x + a)\sinh(x + b)\cos(\alpha) \\
&= \cosh(x + b)\sinh(x + a) + \cosh(x + a)\sinh(x + b) \\
&\qquad - \cos(\alpha)(\cosh(x + b)\sinh(x + a) + \cosh(x + a)\sinh(x + b)) \\
&= (1 - \cos(\alpha))(\cosh(x + b)\sinh(x + a) + \cosh(x + a)\sinh(x + b))
\end{aligned}$$

Since $0 < \alpha \leq \pi$, the term $1 - \cos(\alpha)$ is greater 0. Due to cosh and sinh being greater 0 for $x > 0$, $\cosh(x + b)\sinh(x + a) + \cosh(x + a)\sinh(x + b)$ is positiv as well. Therefore, increasing the radius of p and q by r increases their distance. □

Lemma 2.4. *For every angle $\alpha \in (0, \pi]$ there are radii r, R with $\frac{R}{2} \leq r \leq R$, such that hyperbolic points $p = (r, 0)$ and $q = (r, \alpha)$ have distance R.*

Proof. Recall the hyperbolic distance is defined as $d(p, q) = \operatorname{arcosh}(\cosh(r)\cosh(r) - \sinh(r)\sinh(r)\cos(\alpha))$. We start with $r = \frac{R}{2}$ for an arbitrary but fixed ground space radius $R \in \mathbb{R}$. If the angle is π, the distance $d(p, q)$ is R and we are done. If the angle α is less than π, the distance $d(p, q)$ with this radius is smaller than R. According to Lemma 2.3, if we increase the radius of p and q at the same time, their distance increases. Since we have to stay in the ground space we can only increase the radius to be $r = R$ where the point p, q have maximum distance. Therefore, we set the radius of both points to $r = R$. If this makes $d(p, q)$ to be greater R, there was a radius $r < R$ such that the points p, q had distance R and we are done – recall we started with $d(p, q) < R$ and are now at $R < d(p, q)$, furthermore the distance function when increasing both radii at once is a smooth monotone function , i.e., $\operatorname{arcosh}(\cosh(x + r)^2 - \sinh(x + r)^2\cos(\alpha))$ for variable x.

In the case $d(p, q)$ is still smaller R (with radius of the p and q set to $r = R$), we increase the ground space radius R. To show, that there is a value for R, such that $p = (R, 0)$ and $q = (R, \alpha)$ have distance R, we interpret p and q as corners of an equilateral triangle, with the origin as the third corner and side length R. For hyperbolic equilateral triangles, the relation between the side length s and the angle A is given by $\cosh(\frac{s}{2}) = \frac{1}{2\sin(\frac{A}{2})}$. When we increase the side length s, $\sin(\frac{A}{2})$ has to become smaller. For $s = 0$ the formula yields $A = \frac{\pi}{3}$. For $x \in [0, \frac{\pi}{3})$, $\sin(x)$ monotonously goes to 0, thus the term $\sin(\frac{A}{2})$ can only become smaller when A becomes smaller. Both $\cosh(x)$ and $\sin(y)$ are continuous. For x to infinity $\cosh(x)$ monotonously goes to ∞, thus for $\cosh(\frac{s}{2}) = \frac{1}{2\sin(\frac{A}{2})}$ to hold, A has go to 0. Therefore, if we choose R large enough, we can set the radius of p and q to R and get any angle between $0 < \alpha < \frac{\pi}{3}$. □

Lemma 2.5. *Cycles are SHDGs.*

Proof. Let C_i be the cycle with i nodes. We construct an embedding of an SHDG that represents C_i. To that end, we want to embed the nodes with the same radius and angles $\frac{k \cdot 2\pi}{i}$ for $0 \leq k \leq i-1$, i.e., $\frac{2\pi}{i}$ between nodes. According to Lemma 2.3, there is a radius r for the nodes and a radius R for the embedding, such that nodes with angle $\alpha = \frac{2\pi}{i}$ have distance R. Like this, each node is adjacent to the node with angle $\frac{2\pi}{i}$ larger and smaller and we have an embedding of C_i. □

The next Lemma sounds quite technical, it just states that if we take an embedding of an SHDG, an HDG or an EDG and draw the edges between adjacent nodes as geodesics, a crossing of edges implies three of the four endpoints of the edges to form a cycle.

Lemma 2.6. *Let G be a unit disk graph with fixed embedding e of radius R, and let $e_1 = \{x_1, x_2\}$, $e_2 = \{y_1, y_2\} \in E$ be edges. If the geodesic from x_1 to x_2 crosses the geodesic from y_1 to y_2, then three of the four nodes x_1, x_2, y_1, y_2 induce a C_3.*

Proof. Let i be the intersection of the geodesics. There is a point among x_1, x_2, y_1, y_2, such that the distance to i is minimal. Without loss of generality, let x_1 be this point. The distance from x_1 to y_1 is smaller than or equal to $d(x_1, i) + d(i, y_1)$ due to the triangle inequality. Due to the choice of x_1, $d(x_1, i) \leq d(y_2, i)$, thus $d(x_1, i) + d(i, y_1) \leq d(y_2, i) + d(i, y_1) = d(y_2, y_1) \leq R$. Therefore, x_1 and y_1 are adjacent. The analogous argument applies for x_1 and y_2 to be adjacent and there is a C_3 induced by x_1, y_1, y_2. □

Lemma 2.7. *(Proof form [StE]) For $0 < b \leq a$, with $a, b \in \mathbb{R}$ $\frac{\sinh(b)}{\sinh(a)} \leq \frac{b}{a}$.*

Proof. Let $f(x) = \frac{sinh(x)}{x}$. Since $sinh(x)$ is a non-negative convex function, $f(x)$ is monotonically increasing for $x > 0$. Since $b \leq a$, this shows that:

$$f(b) \leq f(a) \Leftrightarrow \frac{sinh(b)}{b} \leq \frac{sinh(a)}{a} \Leftrightarrow \frac{\sinh(b)}{\sinh(a)} \leq \frac{b}{a}.$$

□

3. On Related Graph Classes

3.1. Euclidean Unit Disk Graphs are HDGs

Let $i = (a, \theta_a)$ and $j = (b, \theta_b)$ be points in the Euclidean or hyperbolic plane with radii a, b and angles θ_a, θ_b. Let $d_e(i,j) = \sqrt{a^2 + b^2 - 2ab \cdot \cos(\Delta_\theta(\theta_a, \theta_b))}$ be the Euclidean distance and let $d_h(i,j) = \operatorname{arcosh}(\cosh(a)\cosh(b) - \sinh(a)\sinh(b)\cos(\Delta_\theta(\theta_a, \theta_b)))$ be the hyperbolic distance, where Δ_θ is the angular distance. We denote the threshold distance by R.

Theorem 3.1. *EDGs are a subclass of HDGs. There is an HDG that is no EDG.*

Proof. We begin by giving an example of a graph that is an SHDG and thus also an HDG but no EDG. Let S_i be the star graph with i leaves. To show it is an SHDG, we take an embedding of the cycle C_{2i} and remove every second node on the cycle we get an independent set of i nodes for which we have an embedding with Lemma 2.5. We embedd an additional node in the origin, to get an SHDG embedding of S_i. But already the star graph S_7 is no EDG [AZ18]. Thus, there is an HDG that is no EDG.

To prove that every EDG is an HDG, we show that for every Euclidean unit disk embedding of an EDG, there is a hyperbolic unit disk embedding. To do so, we look at the hyperbolic plane in native representation, close to the origin. There, hyperbolic distances are almost the same as Euclidean distances. We will see that this can be used to take the Euclidean unit disk embedding, slightly increase the threshold distance and interpret it as hyperbolic embedding while keeping the same edges, i.e., keeping the change in distance small enough to not change any adjacency.

First, we observe that if we increase the threshold distance R, we do not *loose* any edges, i.e., all nodes that were adjacent before still have distance less than or equal to R. For each pair of non-adjacent nodes i and j, let $\varepsilon_{ij} > 0$ be such that i and j would still not be adjacent if the threshold distance was $R \cdot (1 + \varepsilon_{ij})$. Taking the minimum over all pairs $\varepsilon := \min_{i,j} \varepsilon_{ij}$ and enlarging the threshold distance to $R \cdot (1 + \varepsilon)$ will not add any edges.

We use the native representation of the hyperbolic plane. In this representation, hyperbolic distances are never smaller than Euclidean distances. Close to the origin, the hyperbolic plane is similar to the Euclidean plane. The idea of the proof is to show that the hyperbolic distance is at most $1 + \varepsilon$-times the Euclidean distance, for a small radius D, close to the origin. This way, we can directly take the Euclidean unit disk embedding, shrink it and its threshold distance (which we still denote by R) to fit in a disk of radius D. We then interpret the Euclidean polar coordinates of the shrunken embedding as hyperbolic native representation coordinates, of an embedding with threshold distance $R \cdot (1+\varepsilon)$ getting the same adjacencies.

The proof for $d_h \leq d_e \cdot (1+\varepsilon)$ for small D can be found in the next paragraph. First, we show that changing the threshold distance to $R \cdot (1+\varepsilon)$ does not change adjacencies. We look at pairs of nodes i, j. If i and j are adjacent, they have distance $d_e(i,j) \leq R$ in the Euclidean embedding. Since $d_e(i,j) \leq d_h(i,j) \leq d_e(i,j) \cdot (1+\varepsilon) \leq R \cdot (1+\varepsilon)$, setting the threshold distance to $R \cdot (1+\varepsilon)$ will keep the nodes adjacent. If the nodes are not adjacent, $d_e(i,j) > R \cdot (1+\varepsilon)$ in the Euclidean embedding, for ε as defined above. Thus, $d_h(i,j) \geq d_e(i,j) > R \cdot (1+\varepsilon)$ is greater than the new threshold distance, keeping the nodes not adjacent.

Lastly, we want to show $d_h \leq d_e \cdot (1+\varepsilon)$ for small D. To that end, let $i = (a, \theta_a)$ and $j = (b, \theta_b)$ be two hyperbolic points with radii $a, b \leq D$. This choice of a and b allows us to use the following approximation and inequalities for the hyperbolic functions: For x close to 0, $\cosh(x)$ is smaller than or equal to $1 + \frac{x^2}{2} + \mathcal{O}(x^4)$ and $\sinh(x)$ is greater than or equal to x. Thus, $\cosh(a) \cdot \cosh(b)$ is smaller than or equal to $1 + \frac{a^2}{2} + \frac{b^2}{2} + \frac{a^2b^2}{4} + ca^4 + cb^4$ for a constant c and $\sinh(a) \cdot \sinh(b)$ is greater than or equal to ab. Due to the choice of a and b, $1 + \frac{a^2}{2} + \frac{b^2}{2} + \frac{a^2b^2}{4} + ca^4 + cb^4$ is close to 1 and ab is close to 0. Together with the inequality $\operatorname{arcosh}(x) \leq \sqrt{2}\sqrt{x-1}$ for x close to 1, it follows for the hyperbolic distance that:

$$\begin{aligned}
d_h(i,j) &= \operatorname{arcosh}(\cosh(a)\cosh(b) - \sinh(a)\sinh(b)\cos(\Delta_\theta(\theta_a,\theta_b))) \\
&\leq \sqrt{2\cosh(a)\cosh(b) - 2\sinh(a)\sinh(b)\cos(\Delta_\theta(\theta_a,\theta_b)) - 2} \\
&\leq \sqrt{2 + a^2 + b^2 + \frac{a^2b^2}{2} + 2ca^4 + 2cb^4 - 2ab \cdot \cos(\Delta_\theta(\theta_a,\theta_b)) - 2} \\
&\leq \sqrt{a^2 + b^2 - 2ab \cdot \cos(\Delta_\theta(\theta_a,\theta_b))} + \sqrt{\frac{a^2b^2}{2} + 2ca^4 + 2cb^4}.
\end{aligned}$$

Thus, for the ratio of Euclidean and hyperbolic distance,

$$\begin{aligned}
\frac{d_h(i,j)}{d_e(i,j)} &\leq \frac{\sqrt{a^2 + b^2 - 2ab \cdot \cos(\Delta_\theta(\theta_a,\theta_b))} + \sqrt{\frac{a^2b^2}{2} + 2ca^4 + 2cb^4}}{\sqrt{a^2 + b^2 - 2ab \cdot \cos(\Delta_\theta(\theta_a,\theta_b))}} \\
&= 1 + \sqrt{\frac{a^2b^2 + 4ca^4 + 4cb^4}{2a^2 + 2b^2 - 4ab \cdot \cos(\Delta_\theta(\theta_a,\theta_b))}} \text{ holds.}
\end{aligned}$$

We want to shrink the Euclidean embedding to fit into a disk with radius D around the origin, for which the hyperbolic distance is almost the Euclidean distance. To shrink the radii of two nodes in polar coordinates at the same time, we replace each radius r by rx. For $x = 1$, the points still have the same radius as before, to shrink the embedding, we let x go towards 0. In the inequality above, a becomes ax and b becomes bx. Recall, a, b are the fixed radius in the original embedding. Hence,

$$\begin{aligned}
1 + \sqrt{\frac{a^2b^2 + 4ca^4 + 4cb^4}{2a^2 + 2b^2 - 4ab \cdot \cos(\Delta_\theta(\theta_a,\theta_b))}} &\Rightarrow 1 + \sqrt{\frac{a^2b^2x^4 + 4ca^4x^4 + 4cb^4x^4}{2a^2x^2 + 2b^2x^2 - 4abx^2 \cdot \cos(\Delta_\theta(\theta_a,\theta_b))}} \\
&= 1 + \sqrt{\frac{\mathcal{O}(x^4)}{\mathcal{O}(x^2)}} \leq 1 + \varepsilon \text{ for } x \text{ small enough.}
\end{aligned}$$

Thus, if we choose x small enough, $d_h(i,j) \leq d_e(i,j) \cdot (1+\varepsilon)$ holds and every EDG is an HDG as well. □

3.2. On the Relation of SHDGs to HDGs and EDGs

Due to SHDGs being HDGs for which the embedding is limited to a disk of radius R, it is obvious that each SHDG is an HDG as well. In this section, we show that not every EDG and thus not every HDG is an SHDG. To that end, we use an EDG that enforces an independent node to be embedded in a cycle which is not possible in SHDGs. This concept will appear again in the next chapter on recognition. We will show that the graph in Fig. 3.1 is such an EDG that has an independent node drawn in a cycle in every planar drawing of the depicted graph.

The next Lemma states, that there cannot be any independent nodes in a cycle in an SHDG.

Lemma 3.2. *Let G be an SHDG and C be an induced cycle in G. Let e be a fixed embedding of G. If a node v is embedded in the interior of the closed simple curve that connects all nodes of C along the geodesics, then v is adjacent to at least one node in C.*

Proof. Since v lies in the interior of the closed curve that connects all nodes of C along the geodesics, there is a node $c \in C$ that has radius greater than or equal to v as well. C induces a cycle, therefore v is embedded between c and another node of the cycle $c' \in C$ that is adjacent to c, i.e., $\{c, c'\} \in E$. Having radius smaller or equal to c, v is adjacent to c' according to Lemma 2.1. □

In the preliminaries in Lemma 2.6 already, we showed that if we take an embedding of an SHDG, an HDG or an EDG and draw the edges between adjacent nodes along the geodesics, a crossing of edges implies three of the four endpoints of the crossing edges to form a cycle. We use this in combination with Lemma 3.2 to show that the graph in Fig. 3.1 is no SHDG.

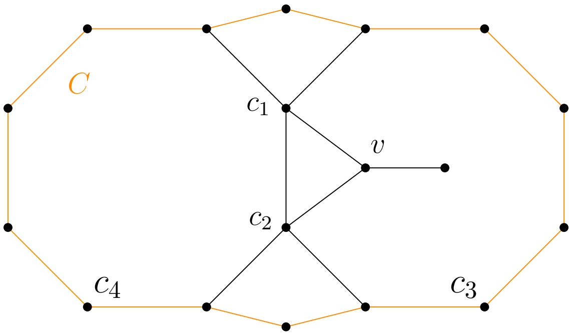

Figure 3.1.: EDG that is no SHDG. In a planar drawing there is always an independent node embedded in a cycle.

Theorem 3.3. *There is an EDG that is no SHDG.*

Proof. Let G be the graph from Fig. 3.1, it is an EDG [BK98]. For the sake of contradiction, we assume that G is an SHDG. As an SHDG it has an embedding e. We consider a drawing of G, where the nodes are embedded according to e and the edges are depicted by the geodesics between the nodes. It has to be a drawing in which no edges cross except for possibly the edges with all of c_1, c_2 and v among their endpoints. This is the case because there would be an induced C_3 otherwise according to Lemma 2.6. Thus, in this drawing, a cycle forms a closed simple curve that connects all nodes of the cycle along the geodesics.

The nodes c_1 and c_2 both have to be either in the interior or the exterior of the closed curve of C since the edge connecting the two would cross the closed curve otherwise. If c_1 and c_2 lie inside the closed curve, v has to be embedded there as well since its edges to c_1 and c_2 would intersect the closed curve otherwise. Being embedded in the closed curve of the cycle C, v has to be adjacent to a node of the cycle, according to Lemma 3.2. This is not the case though, a contradiction.

As depicted in Fig. 3.2, if c_1 and c_2 lie outside of the closed curve of C, they form a new cycle C' that has to have the node c_3 or c_4 embedded in the interior of its closed curve of the purple cycle. This is a contradiction to Lemma 3.2 as well. Since there is no way to draw G without an independent node embedded in one of the closed curves, G is no SHDG. □

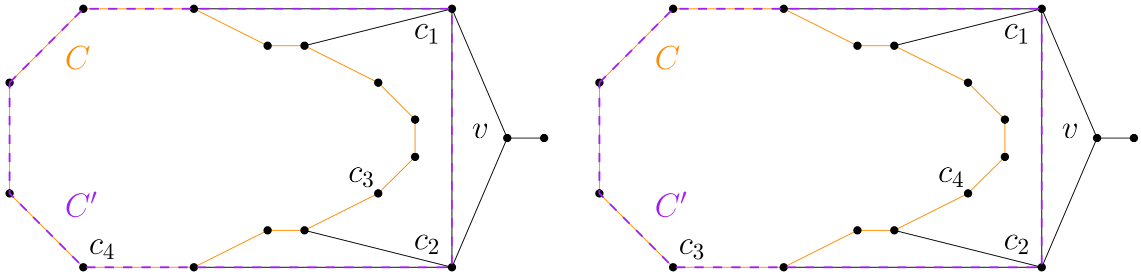

Figure 3.2.: Possibilities of a crossing free drawing of the graph from Fig. 3.1 when c_1 and c_2 are drawn outside the cycle C.

3.3. Chordal Graphs and HDGs are Incomparable

Chordal graphs are graphs in which each cycle of four or more vertices has a *chord*, i.e., an edge connecting two nodes of the cycle that are not neighbors on the cycle. A *perfect elimination scheme* is a total order $\leq$ of the vertices such that the neighborhood of a node v forms a clique in the induced subgraph G_v without the nodes $u \leq v$, i.e., $G_v := G - \{u \mid u \in V, u \leq v, u \neq v\}$. A graph is chordal if and only if it has a perfect elimination scheme.

A *cobipartite elimination scheme* is similar to a perfect elimination scheme. For a total order $\leq$, the neighborhood of a node v forms two cliques in the induced subgraph G_v without the nodes $u \leq v$, i.e., as above $G_v := G - \{u \mid u \in V, u \leq v, u \neq v\}$. SHDGs yield such a cobipartite elimination scheme (cf. Lemma 2.2 and 5.3).

An alternative view on chordal graphs is that of Gavril [Gav74]. For a chordal graph G, there exists a tree T, such that G is the intersection graph of subtrees of T.

In the following, we define the Vapnik-Chervonenkis dimension (VC dimension), informally speaking a measure of how *flexible* a family of sets is. By bounding the VC dimension of hyperbolic disks we show that chordal graphs are not a subclass of HDGs.

Let X be a set called the *ground set* and $\mathcal{R}$ be a family of sets, each of which is a subset of X. Let $Y \subseteq X$ be a subset of X. The *projection* of $\mathcal{R}$ on Y is defined by $\mathcal{R}_{|Y} := \{D \cap Y | D \in \mathcal{R}\}$. We say $\mathcal{R}$ *shatters* Y if $\mathcal{R}_{|Y}$ contains all subsets of Y. The *VC dimension* of $\mathcal{R}$ with respect to X is the maximum cardinality of a subset of X shattered by $\mathcal{R}$.

We are interested in the VC dimension of hyperbolic disks. In this setting, the ground set is the hyperbolic plane, i.e., $X := \mathbb{H}^2$, and $\mathcal{R}$ are hyperbolic disks with arbitrary radius and midpoint. Furthermore, Y is an arbitrary set of hyperbolic points in the plane. To get the VC dimension, we want to find the maximum number of hyperbolic points $|Y|$, such that for each subset $Y' \subseteq Y$ there is a disk that contains exactly the points $y \in Y'$ among all points in Y.

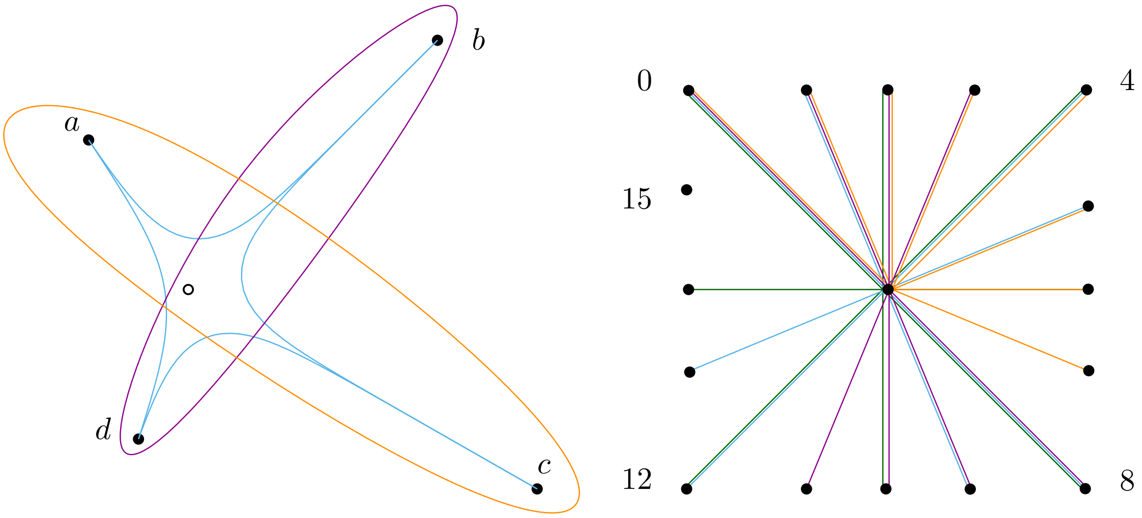

Figure 3.3.: On the left in native representation: The nodes a, b, c, d in convex position with the convex hull in blue. In orange and purple, the schematic representation of two disks containing only a, c (orange) and only d, b (purple), respectively. The disks are shown schematically, since the actual hyperbolic disks would contain more than two nodes. On the right: S_{16} like in Theorem 3.5, with leaves numbered in clockwise order. The subtrees t_1 in orange, t_2 in purple, t_3 in blue and t_4 in green.

Lemma 3.4. *The VC dimension of hyperbolic disks is less than 4.*

Proof. This proof is similar to the one for Euclidean disks by Har-Peled in his lecture notes Example 20.1.1 [HP]. We want to shatter a set P of 4 hyperbolic points in general position by the set of hyperbolic disks. If only 3 points in P are on the boundary of the convex hull of P and one is in the interior, there is no hyperbolic disk that contains only the 3 points on the boundary of the convex hull but not the one in the interior. This is due to hyperbolic disks being convex. Thus, the 4 points $P = \{a, b, c, d\}$ have to all lie on the boundary of the convex hull. Fig. 3.3 on the left illustrates this case. The nodes a, c and b, d are not neighboring on the convex hull. For some disks to shatter P, there has to be a disk only containing the pairs of points a, c and b, d. In Fig. 3.3, these disks are shown schematically in purple and orange. Since the orange disk must not contain b, d and all 4 points are in convex position, the boundaries of the orange and purple disk need to intersect in 4 points. But non-identical hyperbolic disks intersect in two or less points – a contradiction. There are no two disks like the schematically shown orange and purple one only containing a, c or b, d, respectively and hyperbolic disks can not shatter a set of 4 points. □

Theorem 3.5. *The class of chordal graphs and the class of HDGs are incomparable.*

Proof. Cycles of any length are HDGs (cf. Lemma 2.5), thus there are HDGs that are no chordal graphs.

We construct a chordal graph, that is not an HDG. Since a chordal graph is the intersection graph of subtrees of a tree T, it is sufficient to construct a tree T together with some of its subtrees. Let the tree $T = S_{16}$ be the star with one node in the middle and 16 leaves numbered from 0 to 15. In the following, the leaves will represent all possible 2^4 subsets of a set of 4 points. This construction is shown in Fig. 3.3 on the right. The subtrees look as follows: there is one subtree for each leaf, that only consists of the leaf node. Then there are 4 subtrees t_1 to t_4, that all contain the middle node and some of the leaves. The subtree t_1 contains the leaves 0 to 7 but not 8 to 15. The subtree t_2 only contains the leaves 0 to 3 and 8 to 11. The subtree t_3 only contains leaves 0, 1, 4, 5, 8, 9, 12 and 13 and the subtree t_4 only contains the even-numbered leaves. This construction is similar to the construction of a binary counter. While leaf 0 is in all subtrees leaf 1 is only part of t_1 to t_3. This goes all the way down to leaf 15 being part of none of the trees t_1 to t_4. We denote the intersection graph of t_1 to t_4 and the subtrees in the leaves by G.

We now look at HDGs in inclusion representation with radius R. If we want to find a hyperbolic unit disk embedding of the chordal graph G constructed above, we need a disk corresponding to each of the leaves and one corresponding to each of t_1 to t_4. The disks of the leaves need to contain the center of the disks of t_1 to t_4 if they are adjacent to the node of the respective subtree. Thus, if there exists a hyperbolic unit disk embedding of G, it shatters the set of center points of the disks of t_1 to t_4 with the disks of the leaves. This contradicts Lemma 3.4 though, thus G is not an HDG.

Altogether we have seen, that there are HDGs that are no chordal graphs and chordal graphs that are no HDGs, which makes the graph classes incomparable □

Corollary 3.6. *The existence of a cobipartite elimination scheme is a necessary but not sufficient condition for a graph being an SHDG.*

Proof. A perfect elimination scheme is a cobipartite elimination scheme. If the existence of a cobipartite elimination scheme was a sufficient condition for a graph to be an SHDG, each chordal graph would be an SHDG. This contradicts Theorem 3.5 and the existence of a cobipartite elimination scheme is not sufficient for a graph to be an SHDG. □

3.4. Circular Arc Graphs

Looking at SHDGs, some similarities to circular arc graphs (CAGs) can be observed. While CAGs are intersection graphs of circular arcs at the boundary of a disk, SHDGs are intersection graphs of disks within a disk. We can place the disks to all have midpoints with the same radius though. In this section, we use this to show that unit circular arc graphs are SHDGs. We further show that not even the inner neighborhood of an SHDG is a CAG in general, thus showing that there are HDGs and SHDGs that are no CAGs.

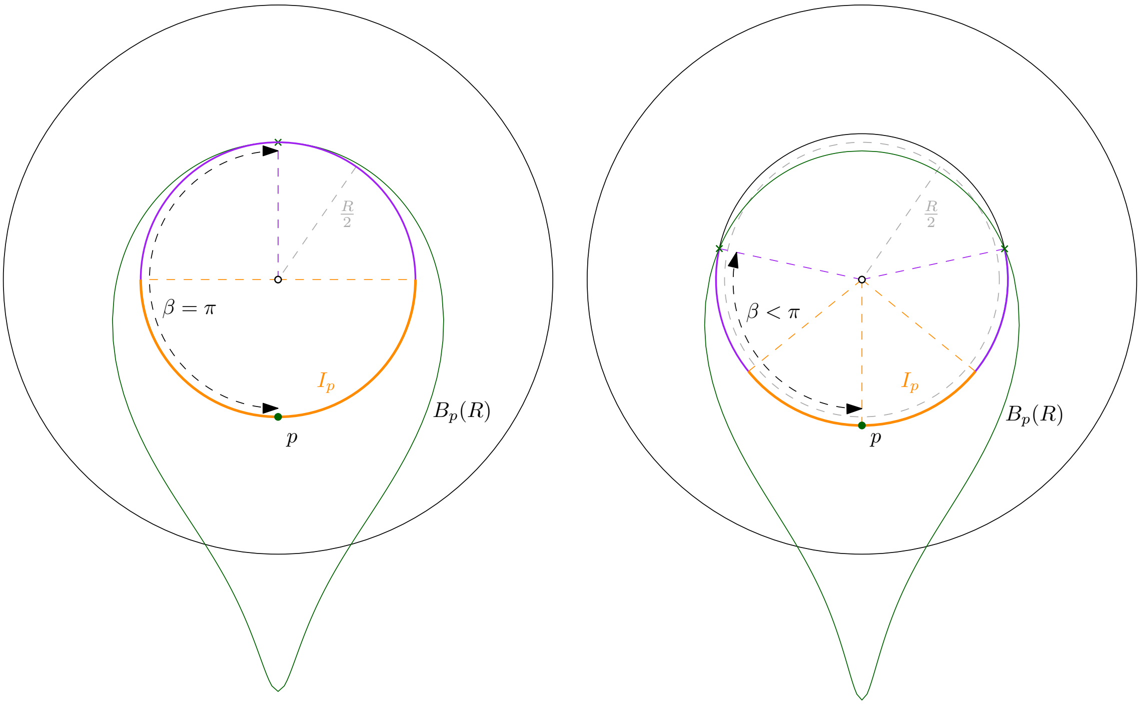

Figure 3.4.: For a point p with radius r, the points with the same radius and distance smaller or equal to R from p form a circular arc. On the left, p has radius $r \leq \frac{R}{2}$. On the right, p has radius $r > \frac{R}{2}$. The angle covered by the arc is 2β. The orange part of the arc that covers an angle of β is used to directly use an arc model of a unit CAG in an SHDG embedding.

Lemma 3.7. *Let p be a hyperbolic point with radius r. The set of points with radius r and distance not larger than R from p forms a circular arc.*

Proof. Let U denote the set of points with the same radius as p with distance not greater than R to p. We distinguish two cases: for case one, let the radius of p be $\frac{R}{2}$ or less. All points with radius $\frac{R}{2}$ or less have distance R or less. The nodes with the same radius as p form a full circle, thus U is this full circle and therefore a circular arc. This is illustrated in Fig. 3.4 on the left. The point $p = (\frac{R}{2}, \frac{3\pi}{2})$ is shown together with the disk $B_p(R)$ around it. The other points with radius $\frac{R}{2}$ are displayed as the orange and purple disk. $B_p(R)$ contains the full gray circle. For case two, let the radius of p be greater than $\frac{R}{2}$, i.e., $r = \frac{R}{2} + c$ for $c > 0$. This is illustrated in Fig. 3.4 on the right. The set U are exactly the points of the circle with radius $\frac{R}{2} + c$ around the origin that are contained in the disk $B_p(R)$. Since non-identical hyperbolic circles intersect at most twice, the set U contains an arc of the circle with radius $\frac{R}{2} + c$ around the origin. □

Theorem 3.8. *Unit circular arc graphs are SHDGs.*

Proof. Let G be a unit circular arc graph with a given arc model. Since all circular arcs are of equal length in the arc model, the ratio of the length of an arc to the circumference of the full circle is fixed. We consider the ratio of arc to full circle as an angle which we denote by α. We say an arc *covers* an angle of α. If α is greater than or equal to π, all nodes are pairwise adjacent and we can embed all nodes of G in the origin to get $G = K_n$. Therefore, we assume α to be smaller than π.

We discuss how we can use the statement of Lemma 3.7 to show that unit circular arc graphs are SHDGs. Let p be a hyperbolic point with radius $r > \frac{R}{2}$ and let U denote the set of points with the same radius as p with distance not greater than R to p. We have a look at Fig. 3.4 on the right again. To better understand what angle around the origin is covered by U, we first split U into two parts. The part from p to one endpoint of the circular arc and the part from p to the other endpoint. The angle around the origin that is covered by these arcs is the same angle for both of them due to symmetry. We denote the angle covered by one of the parts by β, thus U covers an angle of 2β. Observe, that p and the endpoints of the parts have distance R due to the construction. Now, we consider another part of U: we define the arc I_p of a point p as the circular arc around the origin with p in the middle and angle $\frac{\beta}{2}$ to both sides. In Fig. 3.4, this part is displayed as the orange arc of the circle. Since I_p covers angle β like the parts discussed before, the left and right endpoints have distance R. Furthermore, we interpret p as a node, and I_p as its arc. Let v, w be nodes that are embedded with radius r (like p) and let v be adjacent to p while w is not adjacent to p. The node p is adjacent to all nodes embedded in U, i.e., it is adjacent to nodes that have β or less angular distance to p. Therefore, the arc I_v and I_p intersect as they have $\frac{\beta}{2}$ to each side of the respective node. I_w and I_p on the other hand do not intersect as p and w have an angular distance greater than β.

To show that G is an SHDG, u´sing an arc I_p for every node $p \in V$ works if β is the same as the angle α of each arc in the circular arc model. We show that I_p can have any angle smaller π and therefore also α. In Lemma 2.4 we showed, that there is a ground space radius R and a radius $r \leq R$, such that points $p = (r, 0)$ and $q = (r, \alpha)$ have distance R for any angle we choose. We choose the angle to be α and interpret p and q as the endpoints of the arc I_p. Therefore, arcs covering any angle smaller π are possible.

We summarize the full construction: we set α, such that $\frac{\alpha}{2\pi}$ is the ratio of the length of an arc to the circumference of the full circle of a unit arc model of G. Then, using Lemma 2.4, we get a radius R for the ground space and a radius r for the nodes such that nodes with radius r and angular distance α have distance R. For an arc a in the arc model we denote the angle of the midpoint by α_a. For each arc a in the arc model, we embed a node with radius r at angle α_a. From the discussion above we know that this is an SHDG embedding for G. □

Corollary 3.9. *Unit interval graphs and proper interval graphs are SHDGs.*

Proof. Since unit interval graphs are a subclass of unit circular arc graphs, they are SHDGs as well. Additionally, the class of proper interval graphs is the same as the class of unit interval graphs as shown in 1964 already [Rob69]. For a version of the proof that is available on the internet, we refer to the article [Gar07]. Therefore, proper interval graphs are SHDGs as well. □

We continue by showing that not all CAGs are HDGs. To that end, we present an SHDG that can be embedded such that the inner neighborhood of a node is a non-CAG.

Lemma 3.10. *Let G be the graph $\overline{C_6}$ with one additional node v, adjacent to all nodes of $\overline{C_6}$. G is an SHDG and there is an embedding of G in which v has the largest radius among all nodes of G.*

Proof. Fig. 3.5 shows G on the right and an Euclidean unit disk embedding in inclusion representation of G on the left. In the embedding, v has a gray full disk, the full disks of nodes of $\overline{C_6}$ are shown in green, purple and orange with their respective midpoints. The disks of the black nodes are not shown, only their midpoints are displayed. In Theorem 3.1, we showed that EDGs are HDGs. In the proof, the Euclidean embedding is put into hyperbolic space *close* to the origin with *very small* hyperbolic radius of the disks.

We transform the embedding of Fig. 3.5 like in the proof of Theorem 3.1 to an embedding of a HDG with origin o in Fig. 3.5. In the Euclidean embedding, v has the largest radius among the nodes of G. The relative positions of the disks are not changed in the proof, only their radius is slightly altered. Thus, v is still the node with the largest radius among the nodes of G in the hyperbolic embedding.

It is left to show, that G is an SHDG. Due to the origin being in the interior of the disk of v with radius R and this disk being the one furthest away from the origin, all nodes can be embedded with a radius not larger than R. Thus, we have an embedding of an SHDG. □

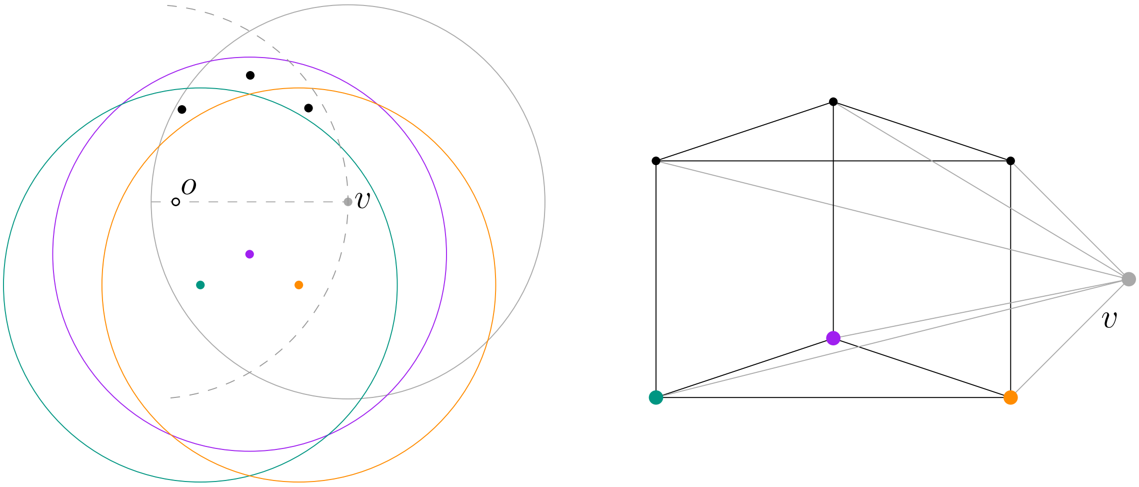

Figure 3.5.: The graph $\overline{C_6}$ with an additional node v that is adjacent to all other nodes. On the left, an EDG embedding of the graph in inclusion representation. On the right the graph itself.

Theorem 3.11. *The inner neighborhood of an SHDG is in general no circular arc graph.*

Proof. Let G be the graph $\overline{C_6}$ with one additional node v, adjacent to all nodes of $\overline{C_6}$ from Lemma 3.10. Since there is an embedding in which v is the node with the greatest radius, it has $\overline{C_6}$ in its inner neighborhood. $\overline{C_6}$ is a well-known minimal non-circular arc graph though, see for example [DGS14]. Thus, neither G nor $G - \{v\}$ are CAGs and the inner neighborhood of an SHDG is in general no CAG. □

Corollary 3.12. *HDGs and SHDGs are no subclass of circular arc graphs.*

Proof. Theorem 3.11 shows that even in an SHDG G the inner neighborhood of a node v must not always be a circular arc graph. Since the inner neighborhood of v is an induced subgraph of G, it is an SHDG and thus an HDG as well. □

4. Complexity of Recognition

4.1. NP-Hardness

In this section, we show that recognizing HDGs is NP-hard. To that end, we use the reduction from Satisfiabiltiy to the problem of recognizing EDGs of Breu *et al.* [BK98] and show that it can be used for HDGs as well. Since EDGs are a subset of HDGs (Theorem 3.1), we only need to show that the gadgets used in the reduction still work the same way in hyperbolic space. Breu *et. al.* use a structure called *k-cages* to transmit the value of a literal to all clauses the literal is a part of. Fig. 4.1 illustrates two *2-cages*. The black cycles are what is called the *cage*, the orange parts are called *beads* and the orange parts in full lines are called the *tips* of the beads. Observe that there is not enough space for the tips of both beads to be embedded in one cycle when they must not intersect. Therefrom, also the name *2-cage* – it is not possible to embed 2 beads in this cage.

More formally, *k-cages* are cycles that prevent k independent tip nodes to be embedded inside of the cycle. The *size* of a k-cage is the number of nodes in the cycle. The nodes of the tips which we call *tip nodes* for now, must not be adjacent to any of the cycle's nodes, nor to any tip nodes of other beads, i.e., they are independent nodes. Thus, from a disk perspective, the cycle forms a cycle of touching or overlapping disks, such that it is not possible to place k disks within the cycle that neither intersect the disks of the cycle nor each other.

Like in Fig. 4.1, when *one* bead is pointing to the right in a sequence of cages with beads in them, *all* beads have to be pointing to the right for them not to intersect. This way, the negative value of a literal can be transmitted along sequences of cages to all clause gadgets the literal is a part of. A clause gadget is a cage as well that assures that not all literals belonging to this clause are negative, i.e., not all beads coming from the literals are embedded in the clause gadget's cage. For more details on this reduction, see [BK98].

The reduction only uses k-cages with k up to 4, thus we only need to show that Euclidean k-cages for $k \leq 4$ are also hyperbolic k-cages. From now on, the nodes of the tips will be called *interior nodes*. We proceed as follows. Let H be the convex hull of the points a cage (i.e., a cycle) is embedded at. For 1 to 4 interior nodes, we want to show that when the cage size is too small, there is no way to embed these interior nodes within the convex hull H, such that the interior nodes have distance $\geq R$ to the nodes of the cage and to the other interior nodes.

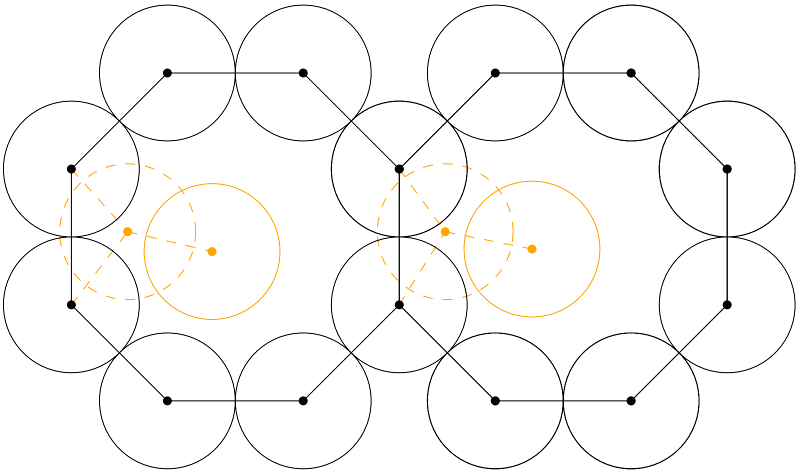

Figure 4.1.: Two cycles of 8 disks, so-called 2-cages with a bead, depicted in orange, in each of them. The full-lined part of the bead is called the tip. For the beads not to intersect, it is not possible to flip the right bead to also be contained in the left cycle.

From a disk perspective, that means, it is not possible to embed the disks of the tips (*interior disks*) without intersecting the disks of the cycle or the other interior disks. To prove this, we consider the case, where the interior disks may touch the disks of the cycle and the other interior disks. For this case, we find lower bounds for how many disks the cycle must consist of. Then we show that when making the interior disks non-touching, it is not possible to embed them inside the cycle of disks anymore.

To get lower bounds for how many disks in the cycle are needed to surround a number of interior disks, we consider all possible embeddings of interior disks that only touch another or are further apart. Each pair of adjacent nodes of the cycle, i.e., each edge, only *covers* a limited *distance* around the interior disks. To measure how much progress in *covering* the interior disks is made by each edge, we define a potential for each embedding of the interior disks. Each edge will be able to cover a potential of 1 or less, therefore when we know the minimum potential over all embeddings of interior disks, we have a lower bound for the number of edges of the cycle and thus for the nodes. We first show the lower bounds, depending on k, for Euclidean k-cages. This is then generalized to also hold for hyperbolic space.

When it comes to the potential, we want an edge of the cycle to cover at most 1 unit of potential. To guarantee this, we split the potential in two parts. Fig. 4.2 illustrates why there are two parts. The figure shows an embedding of a part of the convex hull H of some interior nodes. The segments of the boundary of the convex hull are shown in cyan. When surrounding the convex hull H with edges of the cycle, they, at a time, make progress along exactly one segment on the boundary of H, or around the angle of exactly one point on the boundary of H. An example edge of the cycle is shown in purple. Starting from the left it first makes some distance progress, then it makes some angle progress and finishes by making some more distance progress.

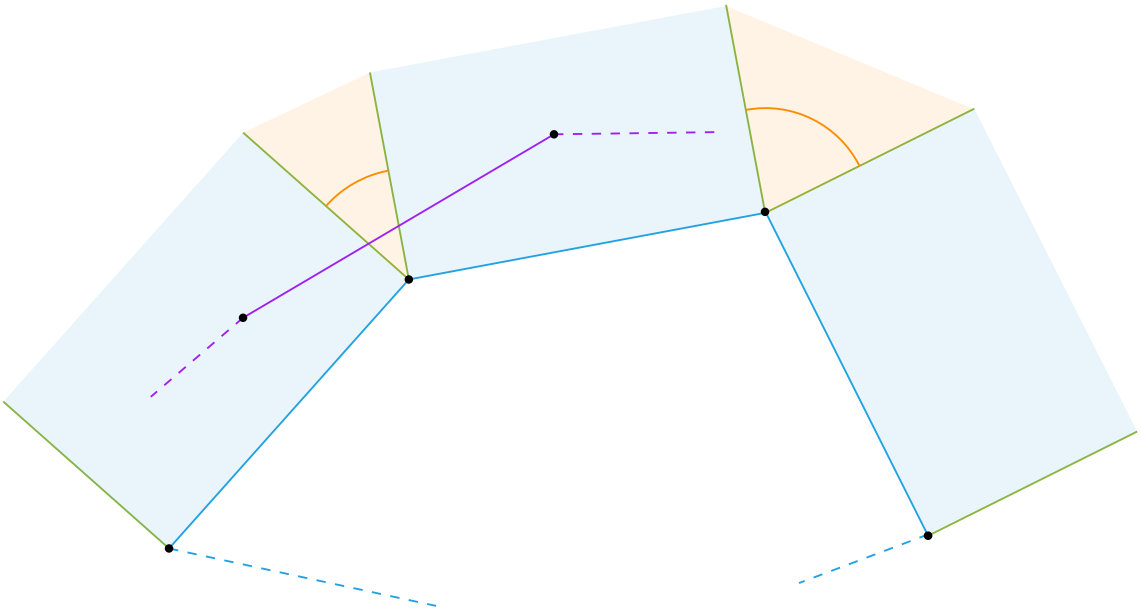

Figure 4.2.: The convex hull of interior nodes with different ways to make progress. Distance potential progress along cyan edges in the cyan areas and angle potential progress around the exterior angle of nodes in the orange area. The purple edge is an example edge of the cycle around the interior nodes.

We call the two types of potential we just saw *angle potential* (the orange part) and a *distance potential* (the cyan part). It follows a definition which is illustrated in Fig. 4.3. The figure is described after the definitions.

Definition 4.1. ***Angle potential***
Given the convex hull H of some interior nodes. Let v be a node on the boundary of H and $\{u, v\}, \{v, w\}$ with $u, w \in H$ be the segments on the boundary of H, v is on. We look at the (not necessarily different) lines g and h through v, perpendicular to the segments $\{u, v\}$ and $\{v, w\}$, respectively. The angle of g and h in v facing away from the convex hull H divided by 60 degree is the angle potential of v. Summing up the angle potential of all nodes on the convex hull yields the angle potential.

Distance potential
The distance potential is the sum of distances between nodes on the boundary of the convex hull H. If all nodes are collinear, the distances are doubled.

The two types of potential are illustrated in Fig. 4.3. The disks are displayed in inclusion representation, there two disks are considered *touching* when the center of one is on the boundary of the other. The orange parts of the disks are the *angle potential* parts, where the angle around an interior node has to be covered by cycle nodes on the orange disk or further away from the node. The green parts are segments of the lines perpendicular to the segments on the boundary of the convex hull of the interior nodes, i.e., the cyan segments. These cyan segments make up the *distance potential* and are covered by cycle nodes between the extension of the green segments. In Euclidean space, the fastest way to cover distance potential is to go parallel to the cyan segments on the boundary of the convex hull, e.g., along the dotted cyan segments.

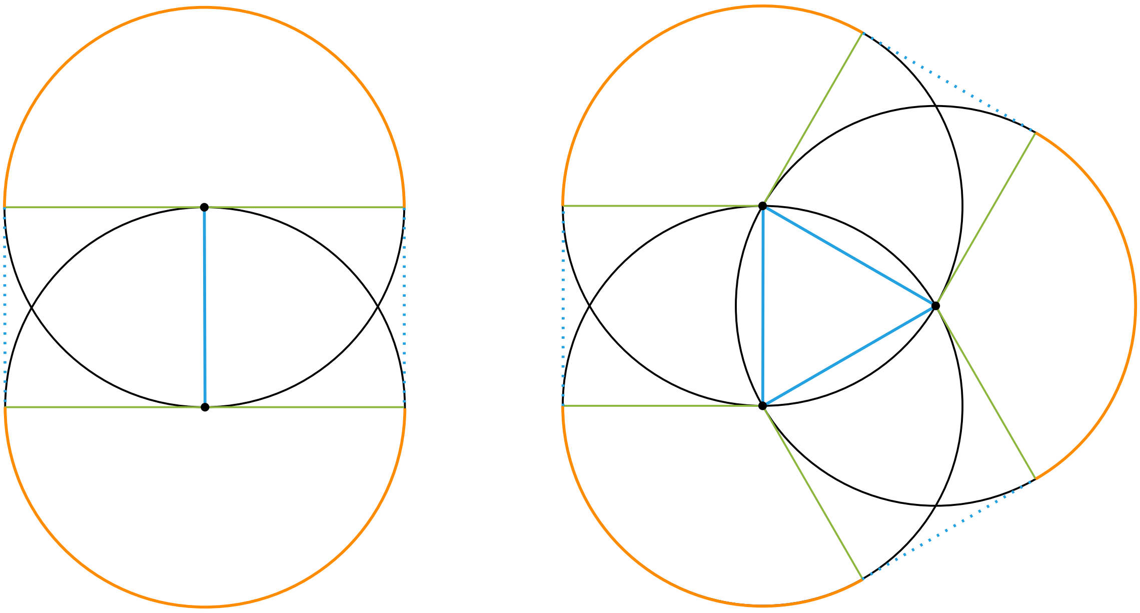

Figure 4.3.: Distance potential in cyan, angle potential in orange. In green, segments of the lines perpendicular to the cyan edges. The dotted cyan segments are possible fastest ways to cover the distance potential in Euclidean space, i.e., tangents.

Lemma 4.2. *The maximum angle potential an edge of the cycle can cover is 1 in Euclidean space and less than 1 in hyperbolic space.*

Proof. Let $e = \{u, v\}$ be an edge of the cycle. The embedding of u and v has to be on the boundary of the interior disks or outside of it. If u and v are on the boundary of *one* disk, the covered angle is maximized. In this case, u and v form a triangle together with the centerpoint of the disk. First, we consider the Euclidean case. There, all sides of the triangle have the same length, thus the angle is 60 degree. If either u or v is not embedded on the boundary, it would make less progress along the disk. Thus, it would cover less angle and the maximum angle potential e can cover is 1.
Now we consider the hyperbolic case. If we take a disk with radius R in native representation around the origin and put the triangle in it, the Euclidean straight segment between u and v already has length R, but hyperbolic segments are no straight lines and consequently longer. This means we can cover less than 60 degree with e in hyperbolic space and the angle potential is less than 1. □

Lemma 4.3. *An edge of the cycle can not cover more than 1 distance potential in Euclidean and hyperbolic space.*

Proof. For the Euclidean case see Fig. 4.4 on the left. We want to make progress along the cyan line. Edges of the cycle are shown as full purple segments of the same length. Here, one edge parallel to the cyan line, one tilted. The green segments are perpendicular to the cyan line and mark how much progress we have made with the purple segments.

An edge parallel to the cyan line covers the distance optimally, i.e., it covers a distance potential of 1. If it is tilted, not the full length of the edge is used to make progress along the cyan line and the distance potential is less than 1.

For the hyperbolic case see Fig 4.4 on the right. The figure uses Poincaré disk representation. Again, we want to make progress along the cyan line. This time, only one edge is shown as

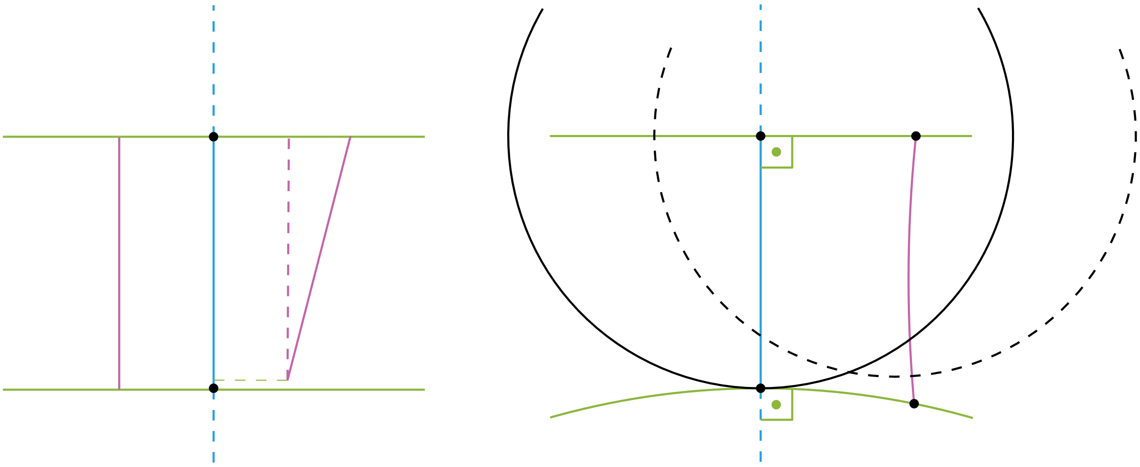

Figure 4.4.: Optimal and suboptimal ways to cover the cyan line in the middle via purple edges. Euclidean on the left, hyperbolic in Poincaré disk representation on the right.

a purple segment. The full black disk is centered at the upper node on the cyan line, the dashed black disk is centered at the upper endpoint of the purple edge. We prove that the progress along the segment connecting the centers, i.e., the cyan line, is always smaller than the length of the edge of the cycle. To measure the progress along the segment connecting the centers, we take the perpendiculars on the segment through the endpoints of the edge, i.e., the green lines. We now want to show that the distance between the intersections of the two perpendiculars with the segment is smaller than the length of the edge, i.e., that the cyan segment is shorter than the purple segment. To do so, we use a Möbius transformation to move one intersection to the origin. The figure shows the situation after the transformation in Poincaré disk representation. Then we take a hyperbolic disk with center on this intersection and the second intersection on the boundary of the disk, i.e., the black disk. If we move this disk along the perpendicular, the disk gets smaller in the Poincaré disk model, because it is moved away from the origin. Additionally, the other perpendicular is no straight line but a circle arc facing away from the first perpendicular. Thus, when we move the black disk all the way to the upper endpoint of the edge it is only the size of the dashed black disk and it does not contain the second endpoint anymore. Thus, the progress made, i.e., the distance potential covered, is smaller than the length of the edge. □

As we already saw in Fig. 4.2, edges of the cycle only cover potential of *one* edge or node at a time. Thus, the potential is a lower bound for the number of nodes needed to have a cycle of disks around a set of interior disks. If the disks of the cycle must not touch the interior disks, we, in particular, need *more* nodes in the cycle than the potential value.

Lemma 4.4. *The angle potential is 6 for any embedding of interior nodes in Euclidean space and more than 6 in hyperbolic space.*

Proof. We start with the Euclidean case. Without loss of generality, the interior nodes are in convex position, the nodes that are not on the boundary of the convex hull have no part in the angle potential. Taking any point in the interior of the convex hull or on its boundary and moving all disks to this point shrinks the distance between all points to 0 but leaves the exterior angles untouched. Thus, the sum of exterior angles is a full circle of 360 degree, i.e., angle potential 6. In the hyperbolic case, the exterior angles get smaller when shrinking the convex hull into one point p like above. To show this, we use a Möbius transformation to move the point p to the origin. Since the segments on the boundary of the convex hull are circular arcs (in Poincaré disk model), the exterior angle of a point is greater than in Euclidean space. Therefore, the sum of exterior angles is more than 360 degree and the angle potential is more than 6. □

Lemma 4.5. *In HDGs, a cycle of 6 is a* 1*-cage, a cycle of 8 is a* 2*-cage, a cycle of 9 is a* 3*-cage and a cycle of 10 is a* 4*-cage.*

Proof at the end of the section.

Theorem 4.6. *Recognizing hyperbolic unit disk graphs is NP-hard.*

Proof. Follows from the reduction in [BK98] together with Lemma 4.5. □

Proof of Lemma 4.5. In all cases, the angle potential is larger than 6 according to Lemma 4.4, we use an angle potential of 6 as a lower bound.

For a single interior disk, there is no distance potential. The angle potential is 6. Thus, a cycle of 6 is a 1-cage.

For 2 interior disks, there are two cases for the distance potential. Case 1, the interior disks touch: the interior disks have distance R and the distance potential is 2, thus the potential is 8. Case 2, the disks do not touch: now, the centers of the disks are further apart and the distance potential is greater than 2. Thus, the potential is greater than 8 and a cycle of 8 is a 2-cage.

For 3 interior disks, there are two cases for the distance potential. Case 1, the centers of the disks are collinear: then the distance potential is 4 if the disks touch or more than 4 if they are further apart. This leads to a potential of 10 or more. Case 2, the centers of the disks are in convex position. If they all pairwise touch, the distance potential is 3. This is minimal since they must not intersect. Thus, the potential is at least 9. Altogether, the potential is 9 or more and a cycle of 9 is a 3-cage.

For 4 interior disks, there are three cases for the distance potential. Case 1, the centers of the disks are collinear: then the distance potential is 6 if the disks touch or more than 6 if they are further apart. This leads to a potential of 12 or more. Case 2, the centers of the disks are in convex position. If the disks of all adjacent nodes on the boundary of the convex hull touch, the distance potential is 4. This is minimal since they must not intersect. Thus, the potential is at least 10. Case 3, the centers of 3 disks are in convex position, one center is in the interior of the convex hull: we can assume that the disks of the nodes in convex position all touch the disk of the node in the interior because we could otherwise simultaneously contract the convex hull to make all distances smaller or keep them the same until all three disks touch the middle disk and possibly each other. This case can be seen in Fig. 4.5. The interior disk with center h is displayed in inclusion representation, thus the centers of the other disks i, j and k (disks are not displayed) *touch* the interior disk. On the left side of the figure, the shortest segment of the boundary of the convex hull is displayed in cyan, the other two segments are displayed in green. We want to show that the sum of these segment lengths is greater or equal to 4. To that end, we first observe that the center of k has to lie between points l and m, the points on the opposite side of i and j on the boundary of the interior disk, otherwise i, j and k would not be the only nodes on the boundary of the convex hull. We furthermore observe that the length of the orange segment connecting i and l is shorter than the green segment between i and k and symmetrically on the other side the orange segment connecting j and m is shorter than the green segment between j and k. This can be seen by considering a disk with center i that has l on its boundary. Both in Euclidean space and in hyperbolic space (in Poincaré disk representation), the disk does not contain k. The same argument holds for the symmetrical case of the segment between j and m. Thus, it is sufficient to show that the sum of lengths of the cyan and the orange segments is 4 or more. In Fig. 4.5 on the right, several triangles are added to the situation. We use the triangles to find the lengths of the colored segments. In Euclidean space, the length of the cyan segment (i.e., the sum of lengths of segments c_1 and c_2 in Fig. 4.5) can be computed by looking at the cyan angle which we denote by α. For right triangles and assuming the radius of the disk around h is 1, the length of c_1 is $\sin(\alpha)$. Since there are two such right-angled triangles, the full length of the cyan segment is $2\sin(\alpha)$. The orange angle together with the cyan angle are a right angle, thus the length of each orange segments (i.e., the segments b_1 and b_2 in Fig. 4.5) is $2\sin(\pi - \alpha) = 2\cos(\alpha)$. We now want to show that the sum of lengths of all colored segments is 4 or more, i.e., $2\sin(\alpha) + 4\cos(\alpha) \geq 4$. To find the minimum, we take the derivative and calculate where it vanishes.

$$\begin{aligned}
& \frac{d}{d\alpha} 2\sin(\alpha) + 4\cos(\alpha) = 2\cos(\alpha) - 4\sin(\alpha) \\
& 2\cos(\alpha) - 4\sin(\alpha) = 0 \\
\iff \quad & 2(\cos(\alpha) - 2\sin(\alpha)) = 0 \\
\iff \quad & \cos(\alpha) - 2\sin(\alpha) = 0 \\
\iff \quad & \cos(\alpha) = 2\sin(\alpha) \\
\iff \quad & \frac{\cos(\alpha)}{\sin(\alpha)} = 2 \text{ for } \alpha \text{ in } (0,\pi] \\
\iff \quad & \cot(\alpha) = 2 \\
\iff \quad & \alpha = \cot^{-1}(2)
\end{aligned}$$

The solutions are $\alpha = \pi n + \cot^{-1}(2)$ for $n \in \mathbb{Z}$. Only the solution for $n = 0$ lies in the relevant interval of angles. We insert this solution in the formula for the sum of lengths: $2\sin(\cot^{-1}(2)) + 4\cos(\cot^{-1}(2)) = 2\sqrt{5}$ which is approximately 4.47, in particular more than 4. The other critical points are at the limits of possible angles, i.e., $\alpha = 0$ and $\alpha = \frac{\pi}{4}$. Inserting this results in: $2\sin(0) + 4\cos(0) = 4$ and $2\sin(\frac{\pi}{4}) + 4\cos(\frac{\pi}{4}) = 3\sqrt{2} \approx 4.24 > 4$.

So far, we have shown that the sum of lengths of colored segments in Fig. 4.5 is greater or equal to 4 in Euclidean space. Now, we have to prove that this still holds in hyperbolic space. To that end, we look at the segments a and b_1 in the figure and show that they are not shorter in hyperbolic geometry. Let l_a and l_{b_1} be the length of the respective segment. From hyperbolic sine law it follows that $\sin(\pi - \alpha) = \sinh(l_{b_1})/\sinh(l_a)$. Lemma 2.7 shows, that $\sinh(l_{b_1})/\sinh(l_a) \leq l_{b_1}/l_a$. Since a has the radius R as its length, rearranging yields $l_{b_1} \geq \sin(\pi - \alpha)l_a = \sin(\pi - \alpha)R$. Thus, b_1 is at least as long in hyperbolic geometry as in Euclidean geometry where we used $R = 1$. With analogous arguments, the length of the cyan segments c_1 and c_2 is greater than or equal to $\sin(\alpha)R$, i.e., their Euclidean length with radius R. Therefore, the length of colored segments in hyperbolic geometry is greater than or equal to $R(2\sin(\alpha) + 4\cos(\alpha)) \geq 4R$. That means the sum of potential is larger than or equal to 10 and a cycle of 10 is a 4-cage. □

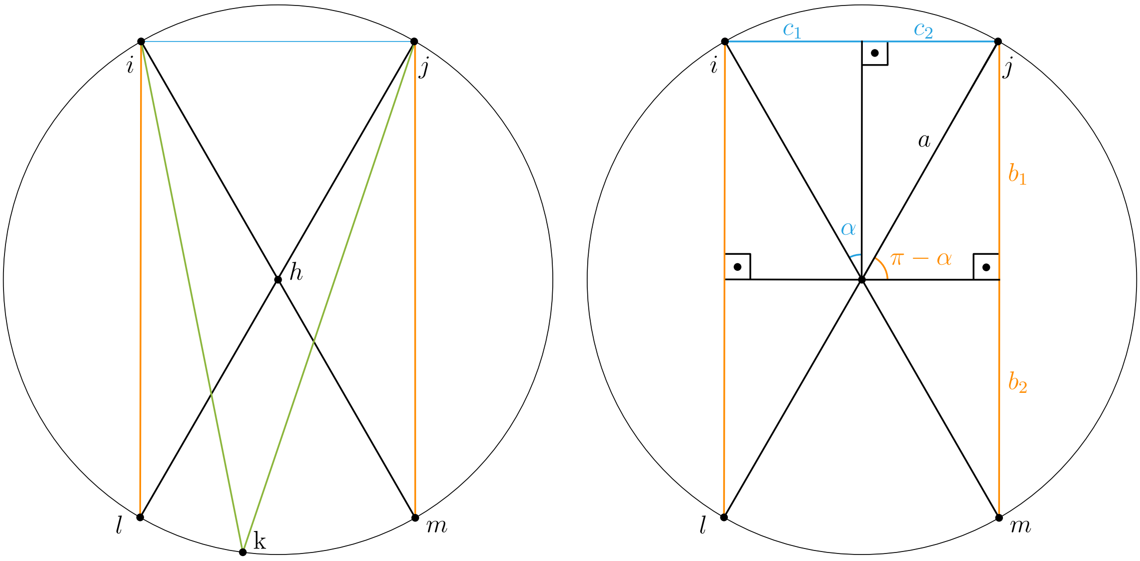

Figure 4.5.: Four interior nodes, three in convex position. The cyan segment is the shortest segment on the boundary of the convex hull. The green segments on the left are the other two segments on the boundary of the convex hull. The orange segments are shorter than the green segments and are used to get a lower bound for the distance potential.

4.2. Membership in $\exists\mathbb{R}$

Let $F(X_1, ..., X_n)$ ba a quantifier-free formula of equalities and inequalities of finitely many real polynomials of finite degree. The *existential theory of the reals* is the set of all true statements of the form $\exists X_1 ... \exists X_n F(X_1, ..., X_n)$. The associated decision problem is to decide for a given statement with formula F whether there exist $X_1, ..., X_n \in \mathbb{R}$ such that $F(X_1, ..., X_n)$ is satisfied.

The complexity class $\exists\mathbb{R}$ is the closure of all true statements of the existential theory of the reals under polynomial-time reduction. A problem is $\exists\mathbb{R}$-*complete* if it is a member of $\exists\mathbb{R}$ and every problem in $\exists\mathbb{R}$ can be reduced to it with a polynomial-time reduction. Recognizing Euclidean unit disk graphs is $\exists\mathbb{R}$-complete [KM12], but there are no results on the complexity of recognizing HDGs, in particular in the context of $\exists\mathbb{R}$. For HDGs, not even membership in $\exists\mathbb{R}$ is clear, since unlike in the Euclidean case where d_e^2 can be described by a polynomial, the formula for d_h we used so far contains multiple hyperbolic functions that must not be used in the statements of the existential theory of the reals. To solve this problem, we introduce another model of the hyperbolic plane, the *hyperboloid model* also known as the *Minkowski model.* It is equivalent to the model we introduced so far but the formula of the distance function is more similar to a polynomial.

In the hyperboloid model, hyperbolic points are represented by a hyperboloid in $\mathbb{R}^3$ with positive first coordinate. The points on this hyperboloid are the points $v = (v_0, v_1, v_2)$ that fulfill $Q(v) = 1$ and $v_0 > 0$ where Q is the *Minkowski quadratic* form $Q(v) = Q(v_0, v_1, v_2) := v_0^2 - v_1^2 - v_2^2$. With the Minkowski quadratic form Q, the *Minkowski bilinear* form B can be defined. For two points on the hyperboloid u, v, $B(u, v) := \frac{Q(u+v)-Q(u)-Q(v)}{2}$. Explicitly this means, $B(u, v) = B((x_0, x_1, x_2), (y_0, y_1, y_2)) = x_0y_0 - x_1y_1 - x_2y_2$. Finally, with the Minkowski bilinear form, the hyperbolic distance is defined by $d(u, v) := \operatorname{arcosh}(B(u, v))$, i.e., the arcosh of a polynomial.

Theorem 4.7. *Recognizing HDGs is a member of $\exists\mathbb{R}$.*

Proof. For a given graph G we construct a quantifier-free formula of equalities and inequalities of real polynomials $F(X_1, ..., X_n)$, such that real numbers $X_1, ..., X_n$ satisfying $F(X_1, ..., X_n)$ exist if and only if the graph G is an HDG. Our approach relies on making the hyperbolic distance function a polynomial. In the hyperboloid model, the formula to calculate hyperbolic distance $d(\cdot, \cdot)$ is no polynomial but contains arcosh. Since arcosh is a monotone function, instead of working with the exact hyperbolic radius and distance we use the Minkowski bilinear form $B(\cdot, \cdot)$ which is a polynomial. The variable X_1 is used as the threshold distance *related* to the actual threshold distance, i.e., if real values $X_1, ..., X_n$ exist such that $F(X_1, ..., X_n)$ is satisfied, it will hold that $X_1 = \cosh(R)$ where R is the actual hyperbolic threshold distance. It follows how we construct the formula F as a conjunction of all equalities and inequalites we add to it. For each node $v \in V$ we take 3 variables $X_{v_0}, X_{v_1}, X_{v_2}$ and add $Q(X_{v_0}, X_{v_1}, X_{v_2}) = 1$ and $X_{v_0} > 0$ to F. Due to these two inequalities for each node, $X_{v_0}, X_{v_1}, X_{v_2}$ are coordinates of a point on the hyperboloid with positive first coordinate. Thus, we have a position in hyperbolic space for each node. If all pairs of adjacent nodes $\{u, v\}$ have distance not greater than R while all non-adjacent nodes $\{x, y\}$ have distance greater R, the necessary and sufficient conditions for edges are fulfilled. For short, we call them *edge conditions.*
For each edge $\{u, v\} \in E$, we add $B((X_{u_0}, X_{u_1}, X_{u_2}), (X_{v_0}, X_{v_1}, X_{v_2})) \leq X_1$ to F and for each non-edge $\{x, y\} \notin E$, $B((X_{x_0}, X_{x_1}, X_{x_2}), (X_{y_0}, X_{y_1}, X_{y_2})) > X_1$ is added. By applying arcosh to both sides of each inequality, they become exactly the edge conditions $\operatorname{arcosh}(B((X_{u_0}, X_{u_1}, X_{u_2}), (X_{v_0}, X_{v_1}, X_{v_2}))) \leq \operatorname{arcosh}(X_1) \Leftrightarrow d(u, v) \leq R$ and $\operatorname{arcosh}(B((X_{x_0}, X_{x_1}, X_{x_2}), (X_{y_0}, X_{y_1}, X_{y_2}))) > \operatorname{arcosh}(X_1) \Leftrightarrow d(x, y) > R$, respectively.

Due to the monotonicity of arcosh, the inequalities with arcosh applied to both sides are equivalent to the inequalities we added to F.

Thus, the construction ensures that if G has an HDG embedding, the edge conditions can be met by hyperbolic points and F is satisfiable. Otherwise, if G is no HDG and therefore has no embedding, the edge conditions cannot be met and F is unsatisfiable.

Since both Q and B are polynomials of constant size and we only use $\mathcal{O}(n)$ variables and add $\mathcal{O}(n^2)$ polynomials to F, the reduction is a polynomial reduction and recognizing HDGs is a member of $\exists\mathbb{R}$. □

5. Maximum Cliques

In this section, we look at maximum cliques in Euclidean and hyperbolic unit disk graphs. Let G be a graph, a clique K is a maximum clique, if there is no clique K' in G which has higher cardinality, i.e., there is no K' with $|K| < |K'|$.

As shown in Lemma 2.2, SHDGs have the property that the inner neighborhood of a node forms two cliques. In a given embedding, any clique has a node the furthest away from the origin. Thus, there is always a node, for which the neighborhood consists of two cliques. We want to use this property to find all maximum cliques. The nodes of a clique in a graph form an independent set in the complement of the graph. Thus, if a graph consists of two cliques, the complement is a bipartite graph. For this reason, we call a graph *cobipartite* if it consists of two cliques. For a node v, we slightly abuse notation and say *the neighborhood $N(v)$ is cobipartite* if $G[N(v)]$ is cobipartite. Abusing notation even more, we say *the node v is cobipartite* if $N(v)$ is cobipartite, i.e., $N(v)$ induces a cobipartite subgraph $G[N(v)]$.

We call an edge the *longest edge* if the two endpoints are the furthest apart among all nodes that are connected by an edge. Similar to the inner neighborhood of a node being cobipartite in SHDGs, in EDGs and HDGs, the neighborhood of the endpoints of the longest edge is cobipartite as well. This has been shown by Clark *et al.* [CCJ90] for EDGs and by Bläsius *et al.* [BFK18] for HDGs. Using this property, they show that finding all maximum cliques can be done in polynomial time if an embedding is given, both in EDGs and HDGs.

Raghavan *et al.* [RS03] showed that in the Euclidean case polynomial time can be accomplished without an embedding. They use an elegant approach of designing their algorithm to be *robust*, that is, it runs on all graphs that have a so-called *cobipartite neighborhood edge elimination ordering* (CNEEO). They claim that checking if a graph admits a CNEEO can be done in polynomial time. This way, they avoid having to check if an input graph is an EDG which is NP-hard, see Chapter 4. There is a gap in one of the arguments of their proof though. This leaves the question whether maximum cliques can efficiently be found unanswered for EDGs.

Our contribution is to generalize the notion of a CNEEO to an elimination ordering which considers the neighborhood of either one or two nodes per elimination step. We further contribute an algorithm that finds all maximum cliques in graphs that admit a *cobipartite neighborhood elimination ordering of nodes* (cf. Definition 5.1) in $\mathcal{O}(\sum_{v \in V} |N(v)|^3)$ and show that SHDGs have such an elimination ordering. The correctness of this algorithm is independent of the correctness of the proof in [RS03]. We contribute evidence that an argument in their proof does not work. For HDGs, our contribution is to show that HDGs admit the necessary *cobipartite neighborhood elimination ordering of edges* (cf. Definition 5.2) such that the algorithm from [RS03] can be applied to find all maximum cliques if the statement in [RS03] is proven to be correct. We further contribute a run time analysis for their algorithm.

To define elimination orderings, we first need some notation. Let $G = (V, E)$ be a graph, for a total order with an ordering of all nodes $L = s_1, ..., s_n$ we define $G_{L,i}$ to be the induced subgraph that does not contain the first $i - 1$ nodes of L, i.e., $G_{L,i} = G - \{s_1, .., s_{i-1}\}$. We further define $N_{L,i}$ to be the neighborhood of the node s_i in $G_{L,i}$.

If $L' = s_1, ..., s_m$ is an ordering of all the edges of G instead, we define $G_{L,i}$ to be the subgraph of G without the first $i - 1$ edges, i.e., $G_{L,i} = (V, E \setminus \{s_1, ..., s_{i-1}\})$. Like above, we define $N_{L,i}$ to be the shared neighborhood of the endpoints of s_i in $G_{L,i}$. Although it is not obvious from the notation of $G_{L,i}$ and $N_{L,i}$ if we use a node ordering or an edge ordering, the notation will only be used when it is clear what kind of ordering L is. We start by generalizing CNEEOs [RS03] to elimination orderings that consider the cobipartite neighborhood of either 1 or 2 nodes. CNEOis with $i \in \{1, 2\}$ (see below) are a generalization because CNEEOs are equivalent to CNEO2s.

Definition 5.1. *An ordering $L = s_1, s_2, ..., s_n$ of all nodes of a graph G is a* cobipartite neighborhood elimination ordering of 1 node *(CNEO1) if $N_{L,k}$ induces a cobipartite graph in $G_{L,k}$ for $1 \leq k \leq n$.*

Definition 5.2. *(Similar to [RS03]). An ordering $L = s_1, s_2, ..., s_m$ of all edges of a graph G is a* cobipartite neighborhood elimination ordering of edges *(CNEO2) if $N_{L,k}$ induces a cobipartite graph in $G_{L,k}$ for $1 \leq k \leq m$.*

We tried to generalize CNEOis for $i > 2$ as well but they do not work for finding maximum cliques the way we use CNEO1s and CNEO2s in this chapter. Furthermore, it is not clear how to define them in the first place, since there is no obvious candidate for elimination when it comes to sets of 3 or more nodes like there is in the case of sets containing single nodes or sets containing two adjacent nodes, i.e., edges.

5.1. Finding Maximum Cliques with Cobipartite Elimination Orderings of Nodes

We now show that SHDGs admit a CNEO1. Afterward, we mainly discuss CNEO1s and only, in the end, apply the findings for them to SHDGs. CNEO2s are mainly discussed in the next section. Since the motivation for using CNEO1s and CNEO2s is the same, we still discuss some of their properties in this section.

Lemma 5.3. *Let G be an SHDG. Then G admits a CNEO1.*

Proof. Since G is an SHDG, it has an embedding in the hyperbolic plane. For this embedding, we can order the nodes from maximum to minimum radius. Let $L = s_1, ..., s_n$ be this ordering where s_1 is one of the nodes with maximum radius. We prove that L is a CNEO1. Assume we have eliminated the first $k - 1$ nodes of L and come to the first elimination, where s_k has no cobipartite neighborhood in $G_{L,k}$. From Lemma 2.2 we know that the inner neighborhood of a node in an SHDG is cobipartite. Due to the ordering of L by the radius of the nodes, s_k only has inner neighbors left in $G_{L,k}$, thus its neighborhood is cobipartite and L is a CNEO1. □

To see how we can use CNEO1s and CNEO2s , we first look at cobipartite graphs. They consist of two cliques, the following Lemma 5.4 shows how to find the maximum cliques among these two.

Lemma 5.4. *Let G be a cobipartite graph of n nodes with complement graph $\overline{G}$. All maximum cliques of G can be computed in $\mathcal{O}(n^{2.5})$.*

Proof. A maximum clique in G is a maximum independent set in $\overline{G}$. Furthermore, the complement of a minimum vertex cover is a maximum independent set. In bipartite graphs, a minimum vertex cover can be computed from a maximum matching in linear time with Kőnigs theorem (see, e.g.,[CCPS09]). Thus, it is sufficient to compute a maximum matching, which is possible in $\mathcal{O}(n^{2.5})$ with the Hopcroft–Karp algorithm [HK73]. From the minimum vertex cover, we get a maximum independent set, from which we get a maximum clique. If the clique has cardinality $\frac{n}{2}$ we have to check if the other $\frac{n}{2}$ nodes form a clique as well. This can be done by checking if they are pairwise adjacent in $\mathcal{O}(n^2)$. □

Lemma 5.4 gives us an efficient way for getting maximum cliques when we are given a cobipartite graph. This leads us to the relevance of a CNEO1 L for finding maximum cliques. For each maximum clique K, there is a first node v of the clique that is eliminated by L. Let i be the index of v in L. At the time of elimination K still has all its edges and nodes in $G_{L,i}$ – we do not eliminate edges and v is the first node of K in L. Therefore, K is in the cobipartite neighborhood $N_{L,i}$ in $G_{L,i}$ and is found using Lemma 5.4.

The same argument holds for a CNEO2 L'. There is always a *first* appearance of an edge e of K in L' with index i, such that the neighborhood of its endpoints in $G_{L,i}$ is cobipartite and the maximum clique K is found using Lemma 5.4.

We have seen so far, how we can use a CNEO1 or CNEO2 to find the maximum cliques. We continue by showing how to find a CNEO1 in the first place. In the next section, we return to discuss CNEO2s as well. From Lemma 5.3 we know that SHDGs always have a CNEO1 and how to find it given an embedding. If no embedding is given, it is not obvious how to find a CNEO1 though. We first present a greedy algorithm that finds a CNEO1 – not only for SHDGs but for all graphs that admit a CNEO1. Then, we combine an improved version of the greedy algorithm with the algorithm to find maximum cliques in cobipartite graphs from Lemma 5.4 to find all maximum cliques in any graph admitting a CNEO1.

The greedy algorithm to find a CNEO1 works as follows: it starts with an empty ordering L. In each iteration for all nodes v that have not yet been added to L it is checked, if their neighborhood induces a cobipartite graph. If so, v is added to L and removed from the graph. The algorithm continues until all nodes have been eliminated.

Lemma 5.5. *Let G be a graph that admits a CNEO1 $L = s_1, ..., s_n$. The greedy algorithm finds a CNEO1 for G in $\mathcal{O}(n^2\Delta^2)$ where Δ is the maximum degree.*

Proof. For the sake of contradiction, assume the greedy algorithm terminates after only having found a partial CNEO1 $L' = s'_1, ..., s'_k$ with $k < n$. Let i be the first index in L of a node s_i in $G_{L',k}$. Furthermore, let N_{s_i} denote the neighborhood of s_i in $G_{L',k}$. Compared to G, in $G_{L,i}$ exactly the nodes s_l with $l < i$ are removed. Due to the choice of i, the same nodes are also removed from $G_{L',k}$ and at least one more. Therefore, the neighborhood of s_i in $G_{L',k}$ is a subset of the neighborhood of s_i in $G_{L,i}$, i.e., $N_{s_i} \subseteq N_{L,i}$. An induced subgraph of a cobipartite graph is a cobipartite graph as well. The neighborhood $N_{L,i}$ induces a cobipartite graph in $G_{L,i}$ by definition of L. Let G^c be this cobipartite graph. Since the neighborhood $N(s_i)$ is a subset of $N_{L,i}$, $N(s_i)$ induces a subgraph of G^c in $G_{L',k}$.

This contradicts the premature termination of the greedy algorithm and thus shows that the greedy algorithm finds a CNEO1 for G.

To find a CNEO1, the greedy algorithm has to check for $\mathcal{O}(n)$ nodes in every iteration if their neighborhood induces a cobipartite graph. The neighborhood of a node v has cardinality of Δ at most. Thus, $\overline{G[N(v)]}$ has less than Δ^2 edges. To find out if a graph is bipartite can be done in linear time using a BFS. Therefore, checking if $\overline{G[N(v)]}$ is bipartite takes $\mathcal{O}(\Delta^2)$ time. Since there are n iterations, we get the desired run time of $\mathcal{O}(n^2\Delta^2)$. □

Corollary 5.6. *Let G be a graph that admits a CNEO1. All maximum cliques can be found in $\mathcal{O}(n^2\Delta^2)$.*

Proof. The algorithm goes as follows. We run the greedy algorithm from Lemma 5.5 to get a CNEO1 $L = s_1, ..., s_n$ in $\mathcal{O}(n^2\Delta^2)$. Then we run the algorithm to get all maximum cliques from Lemma 5.4 on the cobipartite graphs $G_{L,i}[N_{L,i}]$ for i going from 1 to n. We keep track of all cliques of largest cardinality we have found so far and return the maximum cliques in the end. This takes $\mathcal{O}(n\Delta^{2.5})$ for an overall run time of $\mathcal{O}(n^2\Delta^2)$.

As discussed earlier, every maximum clique K has a vertex v that is *first* removed by a CNEO1 L. Let i be the index of v in L. When it is removed, K still has all nodes and edges in the cobipartite subgraph induced by the neighborhood of v in $G_{L,i}$. Thus, every maximum clique is found. □

The algorithm in Corollary 5.6 checks for all nodes if their neighborhood has become cobipartite in every iteration. The next algorithm we present improves upon that by only considering the neighborhoods of just eliminated nodes. Furthermore, we analyze the run time more precisely to get better bounds. The intuition of the algorithm is the following. When removing nodes from a cobipartite graph, it remains cobipartite. We keep track of all nodes with cobipartite neighborhood and eliminate one after the other. Every time we eliminate a node v we only have to check for every neighbor $u \in N(v)$ if its neighborhood has become cobipartite – no other neighborhoods have changed. If the neighborhood of u has become cobipartite we can add it to the set of nodes with cobipartite neighborhood to be eliminated. The order in which we eliminate nodes does not matter, since we just greedily eliminate cobipartite nodes which works according to Lemma 5.5.

Theorem 5.7. *Let $G = (V, E)$ be a graph that admits a CNEO1. All maximum cliques can be found in $\mathcal{O}(\sum_{v \in V} |N(v)|^3)$.*

Proof. Let S denote the set in which we keep track of the nodes that are ready for elimination, i.e., for which the neighborhood induces a cobipartite graph. Initially, S is empty. The algorithm goes as follows. We iterate over all nodes $v \in V$ once and check if their neighborhood induces a cobipartite graph. If $G[N(v)]$ is cobipartite, we add v to S. Then we start removing nodes from S until S is empty. This is done as follows. We pick a node $s \in S$ and find the maximum cliques in $G[N(s)]$ like described in Lemma 5.4. The cliques that had the highest cardinality so far are saved until a clique of bigger cardinality is found.

We then save $N_s := N(s)$ for the next step and remove s from G, i.e., $G := G - \{s\}$. Now, for all nodes that were neighbors of s, we check if their neighborhood now induces a cobipartite graph, i.e., we check for all $u \in N_s$ if $\overline{G[N(u)]}$ is bipartite. If so, u is added to S. When S is empty we output the cliques of maximum cardinality. As discussed earlier, every maximum clique K is found when a node of K is eliminated for the first time.

We continue by showing that the algorithm has the desired run time. The neighborhood of a node v has cardinality $|N(v)|$. Thus, the complement graph it induces, $G' = \overline{G[N(v)]}$, has less than $|N(v)|^2$ edges. To check if a graph is bipartite we can run a BFS in linear time.

The algorithm starts by checking for all nodes if their complement is bipartite resulting in $\mathcal{O}(\sum_{v \in V} |N(v)|^2)$ time for this step. Each node is only eliminated once, thus we get $\mathcal{O}(\sum_{v \in V} |N(v)|^{2.5})$ for finding the maximum cliques in the induced cobipartite graphs using the algorithm from Lemma 5.4. Now we come to the step in which the algorithm checks if any neighbor of a just eliminated node has become cobipartite from the elimination. Every node v only has $|N(v)|$ neighbors that can be eliminated and trigger the check whether $G[N(v)]$ is cobipartite now. Checking if $G[N(v)]$ is cobipartite takes $\mathcal{O}(|N(v)|^2)$. Thus, all checks add $\mathcal{O}(\sum_{v \in V} |N(v)|^3)$ to the run time. Overall, this results in the desired run time of $\mathcal{O}(\sum_{v \in V} |N(v)|^3)$. □

Since SHDGs admit a CNEO1, we can directly apply this algorithm for finding maximum cliques to them.

Corollary 5.8. *Let G be an SHDG. All maximum cliques can be found in $\mathcal{O}(\sum_{v \in V} |N(v)|^3)$.*

Proof. Lemma 5.3 shows that G has a CNEO1. Thus, Theorem 5.7 can be applied and all maximum cliques can be found in $\mathcal{O}(\sum_{v \in V} |N(v)|^3)$. □

5.2. Cobipartite Elimination Orderings of Edges

We now come back to CNEO2s. Similar to the greedy algorithm for CNEO1s, Raghavan *et al.* [RS03] give a greedy algorithm for CNEO2s. The greedy algorithm to find a CNEO2 works as follows: It starts with an empty ordering L of edges. In each iteration, it is checked for all edges $e = \{u, v\}$ that have not yet been added to L if the shared neighborhood of u and v induces a cobipartite graph. If so, e is added to L and removed from the graph. The algorithm continues until all edges have been added to L.

In Lemma 5.5 we showed that if a graph G admits a CNEO1, the greedy algorithm finds it. Raghavan *et al.* [RS03] claims the same for the greedy algorithm and graphs that admit a CNEO2 in their Lemma 2 which in our notation is Conjecture 5.9.

Conjecture 5.9. *(Lemma 2 in [RS03]) Let G be a graph that admits a CNEO2 $L = s_1, ..., s_m$. The greedy algorithm finds a CNEO2 for G.*

An argument in their proof does not work though, making it unclear if CNEO2s can be found efficiently. In the following we show why the proof does not work but not that the statement is wrong, i.e., the greedy algorithm may always find a CNEO2 if one exists.

Their proof is analogous to the proof of Lemma 5.5. Let G be a graph that admits a CNEO2 $L = s_1, ..., s_m$. They assume the greedy algorithm has terminated with only a partial CNEO2 $L' = s'_1, ..., s'_k$. Let s_i be the first edge of L that is still present in $G_{L',k}$. They correctly state that the neighborhood of s_i in $G_{L',k}$ is a subset of the neighborhood of s_i in $G_{L,i}$ because all edges that were eliminated in L were also eliminated in L'. But at this point they claim, that this is sufficient to show that the shared neighborhood of the endpoints of s_i in $G_{L',k}$ is cobipartite. When eliminating edges this argument does not work anymore as we show using the graph depicted in Fig. 5.1 on the left. Observe that both the orange and the purple edge can be eliminated. According to the argument of [RS03], for the orange edge to be ready for elimination, it is sufficient that the shared neighborhood of the endpoints u and v stays the same or is just a subset of what it is in Fig. 5.1 on the left. If we eliminate the purple edge first, the shared neighborhood of u and v stays the same. Thus, it should still be cobipartite. This is not the case though and the orange edge can only be eliminated after at least one other edge is eliminated.

We tried to find a proof for the greedy algorithm to always find a CNEO2 if one exists or to construct an example in which it fails. Since we did not succeed, we cite Lemma 2 from [RS03] as Conjecture 5.9.

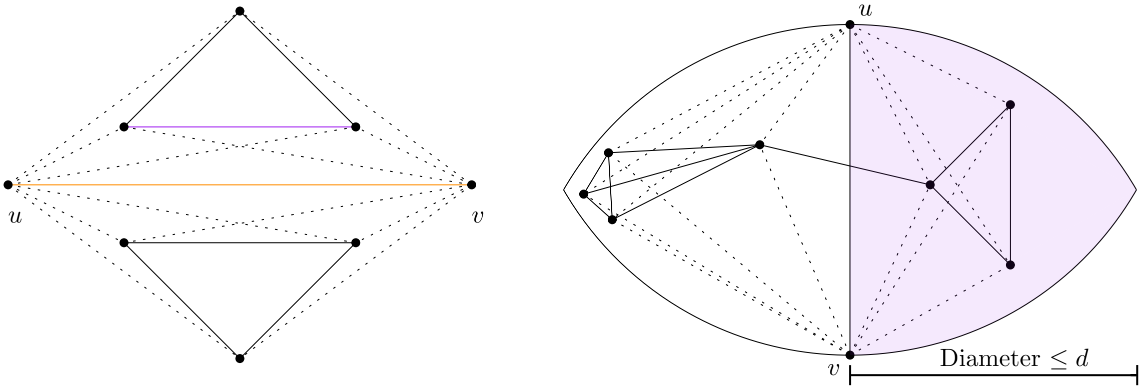

Figure 5.1.: For better readability, less important edges are depicted dotted. On the left: Example for which an argument in the proof of Lemma 2 in [RS03] fails. If the purple edge is eliminated first, the orange edge can only be eliminated after at least one other edge has been eliminated. On the right: The intersection of two disks forming the lens $B_u(d) \cap B_v(d)$. The right half-lens is colored purple. The nodes in each half-lens form a clique.

For the rest of this section, we consider finding maximum cliques in HDGs. We show that in case the greedy algorithm still finds a CNEO2 every time there is one, maximum cliques can efficiently be found in HDGs as well.

Lemma 5.10. *Let $G = (V, E)$ be a graph that admits a CNEO2. If Conjecture 5.9 holds, the greedy algorithm for finding a CNEO2 runs in $\mathcal{O}(m \sum_{\{u,v\} \in E} |N(v)|^2)$.*

Proof. To find a CNEO2, the greedy algorithm in every iteration has to check for $\mathcal{O}(m)$ edges if the shared neighborhood of their endpoints induces a cobipartite graph. The shared neighborhood of an edge $e = \{u, v\}$ has cardinality $|N(v)|$ at most, where without loss of generality $|N(v)| \leq |N(u)|$. Thus, $\overline{G[N(u) \cap N(v)]}$ has less than $|N(v)|^2$ edges. To find out if a graph is bipartite can be done in linear time using a BFS. Therefore, checking if $\overline{G[N(u) \cap N(v)]}$ is bipartite takes $\mathcal{O}(|N(v)|^2)$ time. Since there are m iterations, we get the desired run time of $\mathcal{O}(m \sum_{\{u,v\} \in E} |N(v)|^2)$. □

We take a look back at the algorithm from Corollary 5.6. It finds all maximum cliques by first greedily finding a CNEO1 and then eliminating all nodes. For each elimination, the algorithm from 5.4 to get the maximum cliques is applied to the cobipartite induced graphs.

The same idea works for CNEO2s as well. If Conjecture 5.9 holds, we can efficiently find a CNEO2 and then applying the algorithm from 5.4 for all cobipartite induces subgraphs.

To find maximum cliques in HDGs like this, we only need to show that HDGs admit a CNEO2. To that end, we first cite Lemma 7 in [BFK18] by Bläsius *et al.* that gives us a way to find cobipartite neighborhoods in HDGs.

Lemma 5.11. *(Lemma 7 in [BFK18]). Let $p = (d, 0)$ be the point with radius d and angle 0 in hyperbolic space. Consider a lens of the form $B_0(d) \cap B_p(d)$ in the hyperbolic plane. Then, the half lens $L_1 = \{(r, \varphi) \in L | 0 \leq \varphi < \pi\}$ has geometric diameter at most d.*

Similar to Fig. 2 in [BFK18], Fig. 5.1 on the right illustrates the lens from Lemma 5.11 moved from the origin and p to the points u and v. We discuss what such a lens means for adjacencies in an embedding of an HDG. Let G be an HDG with a fixed embedding and let u and v be the coordinates in the embedding of nodes u' and v' respectively, such that u and v have distance d. Let L be the lens $B_u(d) \cap B_v(d)$. If a node w' is embedded at position w in the lens L, it is adjacent to both u' and v'. This is because by definition of the lens, the distance from u and v to w is smaller than or equal to d. Fig. 5.1 on the right illustrates what the statement of Lemma 5.11 means for us. The nodes placed in each of the half-lenses of L are pairwise adjacent, i.e., they form a clique. With the same idea as in Lemma 5.3, this can be used to show that an HDG has a CNEO2.

Lemma 5.12. *Let G be an HDG. Then G admits a CNEO2.*

Proof. Since G is an HDG, it has an embedding in the hyperbolic plane. We define the length of an edge to be the distance of the two endpoints in the embedding. Using the embedding, we can order the edges from maximum to minimum length. Let $L = s_1, ..., s_m$ be this ordering where s_1 is one of the edges with maximum length. We prove that L is a CNEO2. Assume we have eliminated the first $k - 1$ edges of L and come to the first elimination, where the endpoints of s_k have no cobipartite shared neighborhood in $G_{L,k}$. Let u, v be the endpoints of s_k. All edges longer than s_k have been removed from $G_{L,k}$ already. Thus, the shared neighborhood of u, v in $G_{L,k}$ are exactly the nodes embedded in the lens considered in Lemma 5.11. We know that the nodes embedded in the left and right half-lens form cliques, thus the shared neighborhood of u and v induces a cobipartite graph. Therefore, L is a CNEO2. □

Theorem 5.13. *Let G be an HDG. If Conjecture 5.9 holds, all maximum cliques can be found in $\mathcal{O}(m \sum_{\{u,v\} \in E} |N(v)|^2)$.*

Proof. The algorithm works similar to the one in Corollary 5.6. It goes as follows. We run the greedy algorithm from Lemma 5.10 to get a CNEO2 $L = s_1, ..., s_m$ in $\mathcal{O}(m \sum_{\{u,v\} \in E} |N(v)|^2)$. Then we run the algorithm to get all maximum cliques from Lemma 5.4 on the cobipartite graphs $G_{L,i}[N_{L,i}]$ for i going from 1 to m. We keep track of all cliques of largest cardinality we have found so far and return the maximum cliques in the end. This takes $\mathcal{O}(\sum_{\{u,v\} \in E} |N(v)|^{2.5})$ for an overall run time of $\mathcal{O}(m \sum_{\{u,v\} \in E} |N(v)|^2)$.

As discussed in the previous section, every maximum clique K has an edge e that is first removed by a CNEO2 L. Let i be the index of e in L. When e is removed, K still has all nodes and edges in the cobipartite subgraph induced by the shared neighborhood of the endpoints of e in $G_{L,i}$. Thus, every maximum clique is found. □

6. Separators and Treewidth

Separators and Separator Hierarchies. A *separator* of a graph $G = (V, E)$ is a set $S \subseteq V$ such that the induced subgraph $G - S$ is separated into two components $G[V_1], G[V_2]$ with $V_1 \dot{\cup} V_2 \dot{\cup} S = V$ that have no edges between another, i.e., $\{u, v\} \notin E$ for $u \in V_1$ and $v \in V_2$. The separator is called α-balanced with $\alpha \in (0, 1)$ if $k := \max(|V_1|, |V_2|) \leq \alpha \cdot n$. In a set of separators $\mathcal{M}$ a separator $S \in \mathcal{M}$ is called *most balanced separator* among the separators in $\mathcal{M}$ if there is no separator $S' \in \mathcal{M}$ with components $G[V_1'], G[V_2']$, such that $\max(|V_1'|, |V_2'|) < k$.

A separator hierarchy $(T, \mathcal{S})$ is a rooted binary tree T with t nodes, for which each node is associated to a separator $S_i \in \{S_1, ..., S_t\} =: \mathcal{S}$, such that S_i separates some subgraph $G[V']$ with $V' \subseteq V$ into two components $G[V_{i_1}], G[V_{i_2}]$ with no edges between each other. The root is associated to separator S_1. For node v of the tree T with associated separator S_i, the children have associated separators S_{2i} and S_{2i+1}, respectively. We refer to the nodes of T by their associated separators. The set of separators $\mathcal{S}$ partitions the set of nodes, i.e., $V = \bigcup_{S_i \in \mathcal{S}} S_i$. The root of T, i.e., S_1, is a separator for the whole graph. For a node of the tree S_i that separates $G[V']$ with $V' \subseteq V$ into two components $G[V_{i_1}], G[V_{i_2}]$, the children S_{2i}, S_{2i+1} are separators for $G[V_{i_1}]$ and $G[V_{i_2}]$, respectively. An α-balanced separator hierarchy is a separator hierarchy, for which each node of T is associated to an α-balanced separator.

To give some intuition for separator hierarchies: The root of T separates G into the two components $G[V_{1_1}], G[V_{1_2}]$ that have no edges between another. Thus, the root's children in T, i.e., S_2, S_3, are subsets of V_{1_1} or V_{1_2}, respectively. The same holds for all the descendants of S_2 and S_3 in T. Therefore, the subgraphs induced by S_2 and S_3 ($G[S_2]$ and $G[S_3]$, respectively) are disconnected, as are the subgraphs induced by their respective descendants. Overall, this means that a separator $S_i \in \mathcal{S}$ is separated from all other separators in S except for its ancestors and descendants in T.

Treewidth. Let T be a tree with t nodes, and let $\mathcal{X} = \{x_1, ..., X_t\}$ be a family of sets. Each node i is associated to a set $X_i \in \mathcal{X}$, which is called the *bag* of i. The pair $(T, \mathcal{X})$ is a tree decomposition of a graph $G = (V, E)$ if the bags are subsets of V that have the following two properties.

(i) For each vertex $v \in V$, the nodes of T whose bags contain v induce a subtree of T.

(ii) For each edge $\{u, v\} \in E$, there exist a bag $X \in \mathcal{X}$ with $u \in X$ and $v \in X$.

The width of a tree decomposition is the size of the largest bag minus 1. The *treewidth* $\text{tw}(G)$ of a graph G is the smallest k for which G has a tree decomposition of width k.

The treewidth of a graph is an important parameter. Informally speaking it quantifies how similar to a tree a graph is. When the treewidth is small, many NP-hard graph problems can be solved efficiently. This often works by first finding a tree decomposition for the graph and then using dynamic programming on this tree decomposition. It is important for the efficiency that the tree decomposition has a small width, optimally the treewidth. Tree decompositions are closely related to balanced separators, therefore we begin this chapter by looking at separators and proceed toward treewidth later.

In Section 6.1 we study separators that consist of the neighborhood of two endpoints of an edge. We present an algorithm how to find a $\frac{1}{2}$-balanced separator among these separators and show how it can be used to get a $\frac{1}{2}$-balanced separator hierarchy. After that, we make use of the close relationship between balanced separators and tree decompositions to find a tree decomposition with the $\frac{1}{2}$-balanced separator hierarchy from our algorithm.

In Section 6.2 we discuss grids in SHDGs. Grids are related to treewidth in that every graph of a certain treewidth also contains a grid of a certain size as a minor. We conjecture that while keeping the clique number constant, an SHDG can have a grid minor of any size. This comes at the cost of also having exponentially many nodes though.

6.1. Bounding the Treewidth by the Maximum Degree

In SHDGs, nodes with certain radii have special properties. Therefore, we group the nodes by their radius in two regions. Fig. 6.1 on the left shows the regions we consider together with some nodes and their disks. Let R be the radius of the ground space, the nodes are embedded in. We call the nodes in the blue region with a radius not greater than $\frac{R}{2}$ the *central nodes*. Since their distance to the origin is smaller than or equal to $\frac{R}{2}$, they all have a pairwise distance not greater than R and form a clique. In Fig. 6.1 on the left, the disks are shown in intersection representation with radius $\frac{R}{2}$, thus they all have the origin in common. The white region is called the *rim*, with all nodes of radius greater $\frac{R}{2}$ being called the *rim nodes*. Since the distance of each such node v to the origin is greater $\frac{R}{2}$, the disk of v with radius $\frac{R}{2}$ intersects both the blue region and the boundary of the ground space, all nodes are embedded in.

Before we show how edges between rim nodes induce separators, we first define the *cone* that two nodes form. Let G be an SHDG and let u and v be two nodes that, without loss of generality have angle zero and $\alpha \leq \pi$, respectively. The *interior cone* is the region of the groud space with an angle between 0 and α. The *exterior cone* is the region of the ground space with an angle outside of the range 0 to α. The cones of an edge are the cones of its two endpoints. In Fig. 6.1 on the right, the interior cone of the nodes u and v is shown in orange. Let x be another node of G, we say x is *in the interior/exterior cone of* u, v, if there is an embedding of G in which x is embedded in the interior/exterior cone of u, v. If we want to talk about all nodes in a cone, we denote by *the interior/exterior nodes of* u, v all nodes in the interior/exterior cone of u, v.

Lemma 6.1. *Let G be an SHDG, and let $\{u, v\}$ be an edge between rim nodes u and v. Then $N(u) \cup N(v)$ separates the interior and exterior nodes of u, v.*

Proof. Let the radii of u and v be r_u, r_v, respectively and let $r = \max(r_u, r_v)$ be the larger of the two radii. For rim nodes, the disks with radius $\frac{R}{2}$ intersect both, the region of the central nodes and the boundary of the ground space. Additionally, the two nodes u, v are endpoints of an edge, which means the two disks intersect. Fig. 6.1 on the right illustrates

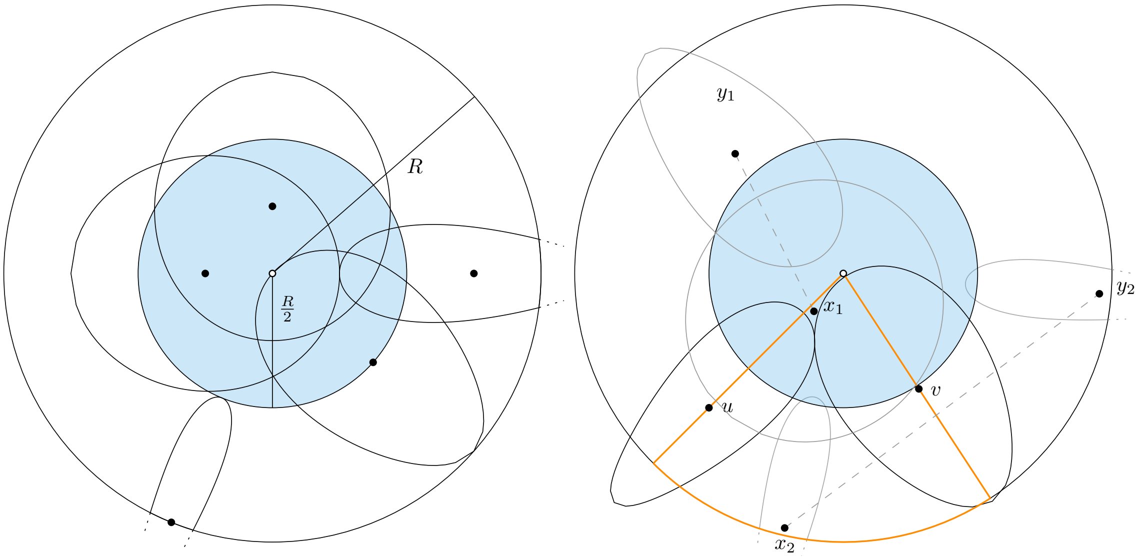

Figure 6.1.: On the left: Region of central nodes in blue and rim in white. Central nodes form a clique, their disks in intersection representation all contain the origin. The disks of rim nodes intersect both the region of the central nodes and the ground space. On the right: The rim nodes u, v, with their interior cone in orange and their disks with radius $\frac{R}{2}$. Nodes x_1, x_2 in the interior cone are separated from nodes in the exterior cone y_1, y_2 by either being adjacent to u or v or not being adjacent to any node in the exterior cone in the first place.

the two disks and their interior cone in orange. We want to show, that the neighborhood $N(u) \cup N(v)$ separates the nodes in the interior cone from the nodes in the exterior cone. To that end, assume there is a node x in the interior cone, adjacent to a node y in the exterior cone with both $x, y \notin N(u) \cup N(v)$. We distinguish two cases.

We first look at a node x with radius $r_x < r$ in the interior cone. Since x lies in between u and v and has smaller radius, x is adjacent to both u and v according to Lemma 2.1. This is a contradiction to $x \notin N(u) \cup N(v)$. In Fig. 6.1, this is illustrated by the node x_1.

In case two, the radius of x is greater than or equal to the radius of u and v, $r_x \geq r$. In Fig. 6.1, only x_2 can be chosen as x. If there is an edge $\{x, y\}$ from x to y, their disks have to intersect another but since $x, y \notin N(u) \cup N(v)$, they must not intersect the disks of u and v. Since the disk of x lies in the enclosed region between the disks of u and v and the ground space, it cannot intersect the disk of y which is not in the enclosed region without also intersecting the disk of u or v. Therefore, no such edge $\{x, y\}$ exists.

Thus, all nodes in the interior cone are not adjacent to any node in the exterior cone or contained in the neighborhood of u or v. □

In the upcoming proofs we need the notion of an edge $\{x, y\}$ *skipping* a node m with respect to some cone T of two nodes u, v. Skipping means, that m is not adjacent to x nor to y and the cone T is contained in one cone of x, y while m is contained in the other. Informally speaking, the edge $\{x, y\}$ connects the nodes to both sides of m and also to both sides of T, making m less relevant when looking for a separator of the leftover node after removing T. In Fig. 6.2 on the left node m is skipped by $\{x, y\}$ with respect to the cone of u_1 and v_1.

As a special case of *skipping* a node, for an edge $e := \{x, y\}$ and a node m in the interior cone of x, y we say that e *crosses* m, when m is not adjacent to either x or y. This is a

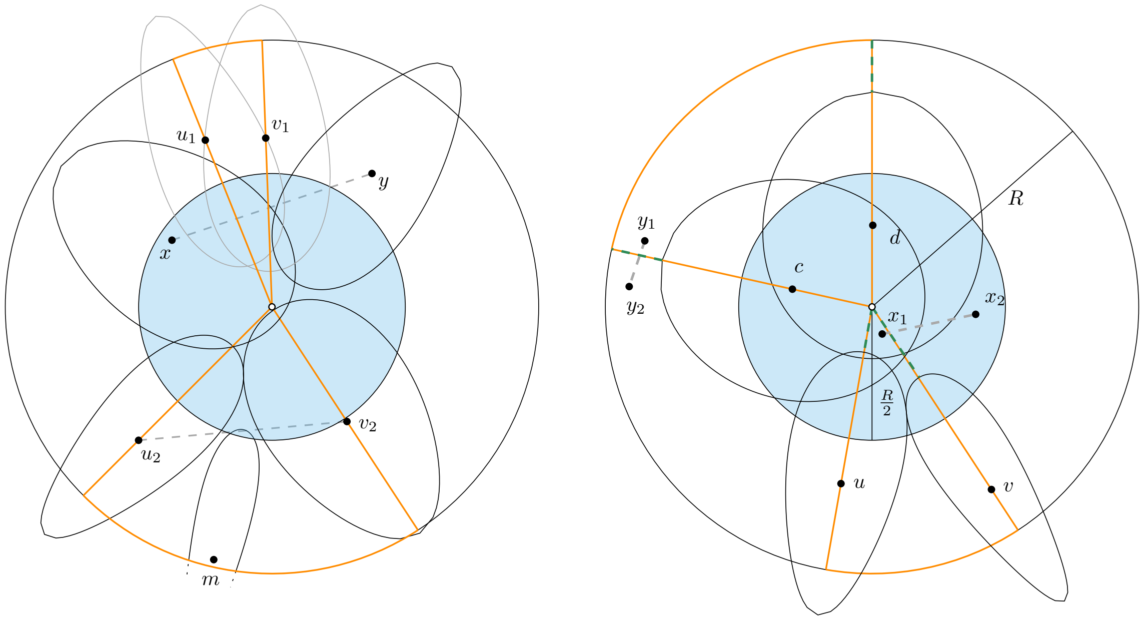

Figure 6.2.: On the left: A node m that is skipped by the edge $\{x, y\}$ with respect to the cone of u_1, v_1 and also crossed by the edge $\{u_2, v_2\}$. On the right: Similarity between rim and central nodes when it comes to their separator properties. Although the disks of the nodes c, d and x, y do not go all the way to the green dashed parts of their cones, the edges $\{y_1, y_2\}$ and $\{x_1, x_2\}$ cannot cross the nodes v and c, respectively. A node of each of the edges is adjacent to v or c, respectively.

special case of a skip, since m is skipped by e with respect to the exterior cone of x, y. Informally that means, $\{x, y\}$ skips over m on the same side of the ground space as m. This is because m is in the interior cone of the other two nodes. In Fig. 6.2 on the left node m is crossed by $\{u_2, v_2\}$.

Corollary 6.2. *Let c be a central node. There is no edge crossing c.*

Proof. Assume there is an edge $\{u, v\}$ crossing c. Both u and v have to be rim nodes since they would be adjacent to c otherwise and thus $\{u, v\}$ would not cross c. Therefore, c is in the interior cone of the two rim nodes u, v and has radius smaller than both of them. Thus, Lemma 2.1 can be applied like in the previous Lemma and c is adjacent to both u and v and $\{u, v\}$ does not cross c. □

The statement of Lemma 6.1 and Corollary 6.2 is illustrated in Fig. 6.4 on the right. The figure shows how central nodes c, d are similar to rim nodes u, v when it comes to their properties of separating the interior nodes from the exterior nodes. The disks of the nodes c, d and x, y look like they can be crossed below and above, respectively, but in both cases the node in the interior cone (x_1 and y_1, respectively) is already contained in the respective neighborhood $N(u) \cup N(v)$ or $N(c) \cup N(d)$. This even holds, if the edge forming the cone has one rim and one central node as endpoint:

Corollary 6.3. *Let G be an SHDG and let $\{u, v\}$ be an edge. Then the neighborhood of $N(u) \cup N(v)$ separates the interior and exterior nodes of u and v.*

Proof. For u, v rim nodes, this follows from Lemma 6.1. For u, v central nodes this follows from Corollary 6.2 – no edge with one node in the interior cone and the other in the exterior cone can cross u or v. For u rim node and v central node it follows from a combination of the above Lemma and Corollary, no edge can cross v, thus u has to be crossed by an edge $\{x, y\}$ between two rim nodes. But the disks of x or y would intersect the disks of u or v like in the proof of Lemma 6.1, since x is enclosed by the disks of u and v. □

The previous corollary shows, that we can separate G by removing the neighborhood of any two adjacent nodes, but we do not yet know if the separator is balanced.

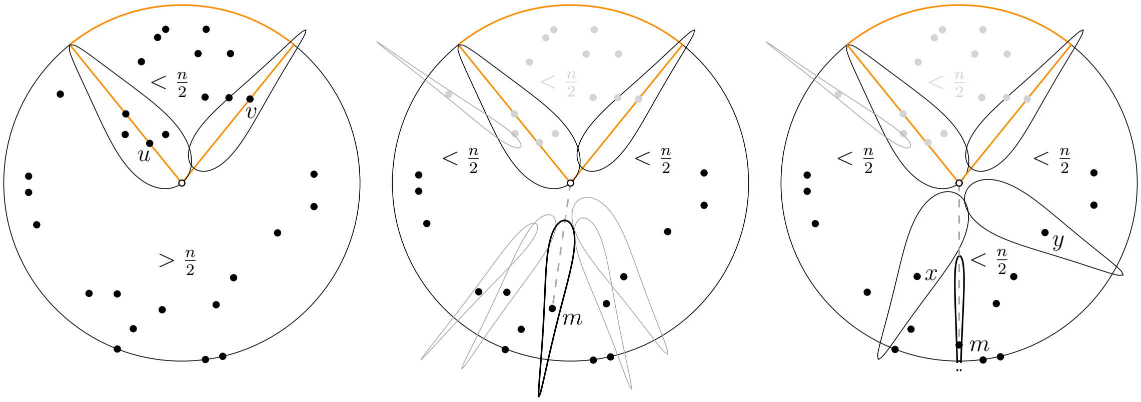

Figure 6.3.: Idea of the proof of Theorem 6.4. On the left, the most balanced separator among the union of neighborhoods of the endpoints of an edge is shown and removed together with the interior nodes of the endpoints. In the middle, case one of the proof: the neighborhood of a node m separates the leftover nodes in a balanced way because m is not skipped. On the right, case two of the proof: m is skipped by an edge $\{x, y\}$. In this case $N(x) \cup N(v)$ completes the $\frac{1}{2}$-balanced separator.

Theorem 6.4. *For an SHDG G there is a $\frac{1}{2}$-balanced separator of cardinality 4Δ or less that consist of the neighborhood of the endpoints of two edges.*

Proof. The idea of this proof is that we first want to find an edge such that its interior cone contains close to $\frac{n}{2}$ nodes and so does its exterior cone. We then remove the neighborhoods of the endpoints of the edge and the nodes in the cone that contains fewer nodes after the removal of the neighborhoods. In general, there is no $\frac{1}{2}$-balanced separator that is just the union of neighborhoods of the endpoints of one edge, thus we are not done yet. We will see that there is a single node or the endpoints of one edge, such that the neighborhoods of these nodes complete our separator to a $\frac{1}{2}$-balanced separator. To find it, we look for an edge (or possibly a single node) for which the neighborhood of the endpoints separates the leftover nodes in components of cardinality $\frac{n}{2}$ or less in its interior and exterior cone. For an illustration of the idea of the proof see Fig. 6.3. On the left, the first edge is chosen that is removed together with the neighborhood of its endpoints and the nodes in its interior cone (because there are fewer nodes in the interior cone than in the exterior). In the middle, the first case of the proof below is illustrated. If a single node m has half of the remaining nodes to each side and there is no edge skipping this node, it is sufficient to remove this node with its neighborhood. On the right, the second case of the proof below is illustrated. If the node m is skipped by an edge with respect to the cone that was already removed, the endpoints of the edge together with their neighborhoods are sufficient to complete the $\frac{1}{2}$-balanced separator.

We continue with the proof. As we saw in Lemma 6.3, the neighborhood of the endpoints of an edge separates the graph. We want to separate G into two components of size $\frac{n}{2}$ or less. To that end, we are looking for a most balanced separator among the neighborhoods of the endpoints of an edge in G. If both components contain $\frac{n}{2}$ nodes or less we are done. Therefore, assume there is one cone that still contains a component C with more than $\frac{n}{2}$ nodes. The number of nodes of C is denoted by n'. Let e be an embedding of G, that we restrict to the larger component C, i.e., we look at $e_{|C}$. For this embedding $e_{|C}$, we define a middle node m. That is, starting from where T was, we look at the nodes by their angle in clockwise order, m is the $\frac{n'}{2}th$ node. Informally speaking, m has half of the remaining nodes on each side. If $N(m)$ already separates the $\frac{n'}{2}$ nodes from the rest of C we are done. If $N(m)$ does not separate C in a balanced way, there is an edge $\{x, y\}$ skipping m with respect to T. We choose an edge $\{x, y\}$ of C, such that $N(x) \cup N(y)$ is the most balanced separator for the whole graph G, among the separators induced by edges in C. Thus, if $N(x) \cup N(y)$ separates C such that all components have less than or equal to $\frac{n}{2}$ nodes, we are done. In the following we show that this is the case, because $\{x, y\}$ would have been chosen instead of $\{u, v\}$ when removing a most balanced separator for the first time. For the sake of contradiction, we assume there is a component C^* with more than $\frac{n}{2}$ nodes. Since $N(u) \cup N(v)$ was the most balanced separator induced by an edge, there is no separator with fewer nodes in a component that still has more than $\frac{n}{2}$ nodes. But if there are more than $\frac{n}{2}$ nodes in C^*, $N(x) \cup N(y)$ is exactly such a contradictory separator since it $C^* \subset C$ but it still has more than $\frac{n}{2}$ nodes. It should be noted, that this argument only works because C^* would be separated from u and v by $N(x) \cup N(y)$ in G already, i.e., C^* would have the same cardinality if it was the component separated by $N(x) \cup N(y)$ in G instead of in C.

Let $S = \{u, v, x, y\}$ be the endpoints of the edges we chose (in some cases S contains m with an adjacent node instead of x, y). Thus, S contains the endpoints of two edges and the union of neighborhoods, which is a $\frac{1}{2}$-balanced separator that has cardinality 4Δ. □

The previous theorem showed that the neighborhood of the endpoints of 2 edges is sufficient to be a $\frac{1}{2}$-balanced separator. But how do we find such a separator? If we are just interested in a simple way of finding such a separator, the naive algorithm runs in polynomial time.

Corollary 6.5. *For an SHDG G a $\frac{1}{2}$-balanced separator of cardinality 4Δ or less can be found in $\mathcal{O}(m^3)$.*

Proof. For each of the $\mathcal{O}(m^2)$ subsets of 2 edges $S \subset E, |S| = 2$, we remove the neighborhoods of the endpoints of all edges in S from G and check if the connected components are of size at most $\frac{n}{2}$. This can be done in $\mathcal{O}(n + m)$, for example via DFS. The existence of such a separator follows from Theorem 6.4. □

From the proof of Theorem 6.4 we can improve upon the run time of Corollary 6.5 with Algorithm 6.1. Since the first *most balanced separator* and the separator of the rest do not have to be chosen at the same time but can be chosen one after the other, there is the possibility to reduce the run time by first choosing the first separator and then choosing the second one.

Theorem 6.6. *For an SHDG G a $\frac{1}{2}$-balanced separator of cardinality 4Δ or less can be found in $\mathcal{O}(m^2)$.*

Proof. The idea of the algorithm is very similar to the idea of the proof of Theorem 6.4. All edges are checked, if the neighborhood of their endpoints separates G such that the largest connected component has $\frac{n}{2}$ or less nodes. This can be done by removing the neighborhoods and running a DFS to find the largest connected component. Then, the

Algorithm 6.1: DegreeSeparate

```
Input: Graph $G = (V, E), n := |V|$
Data: k Cardinality of biggest leftover component
Output: $\frac{1}{2}$-balanced separator $N(\mathsf{S})$ of cardinality $|N(\mathsf{S})| \in \mathcal{O}(\Delta)$
// Initialization
$\mathsf{k} \leftarrow \infty$
$u', v'$ // Nodes for which the neighborhood is the most balanced
    separator so far
$\mathsf{S} \leftarrow \emptyset$

Function Main():
    checkAllEdges ()
    checkAllEdges ()

Function checkAllEdges():
    forall $\{v, w\} \in E$ do
        $G' \leftarrow G - N(v) \cup N(w)$ // $G'$ is the induced subgraph by removing
            the neighborhoods
        $k' \leftarrow G'$.DFS() // get cardinality of biggest connected
            component

        if $k' \leq \mathsf{k}$ then
            $v' \leftarrow v$
            $w' \leftarrow w$
            $\mathsf{k} \leftarrow k'$
    $\mathsf{S} \leftarrow \mathsf{S} \cup \{v', w'\}$
    $G \leftarrow G - N(S)$
```

most balanced separator, i.e., the neighborhood of the endpoints of an edge, for which the largest connected component contained the least number of nodes, is removed from G. We consider the induced subgraph G^* that we get from G by removing the separator. Now, all edges of G^* are checked if the neighborhood of their endpoints separates G^* such that the largest connected component has $\frac{n}{2}$ or less nodes. According to Theorem 6.4 this procedure always finds a $\frac{1}{2}$-balanced separator. It consists of the neighborhoods of 4 nodes, thus the cardinality is 4Δ or less. The run time comes from running a DFS in $\mathcal{O}(m + n)$ for all m edges.

Pseudocode for the procedure above can be found in Algorithm 6.1.
The function checkAllEdges() checks for all edges, if the neighborhood of their endpoints separates G such that the largest connected component is the smallest among all edges, i.e., it finds the most balanced separator among the neighborhoods of endpoints of all considered edges. The first time checkAllEdges() is called, it finds a most balanced separator induced by an edge in G and removes it. Now there is at most one connected component left with more than $\frac{n}{2}$ nodes. The second time checkAllEdges() is called, a most balanced separator induced by an edge of the leftover graph is removed. It is an edge of the biggest leftover connected component since the most balanced separator minimizes the cardinality of the connected component with the most nodes. □

If the embedding of G is also given, a balanced separator can be found faster. To that end, we look at another proof of there being a $\frac{1}{2}$-balanced separator of cardinality $\mathcal{O}(\Delta)$.

Theorem 6.7. *For an SHDG G, a $\frac{1}{2}$-balanced separator of cardinality $\mathcal{O}(\Delta)$ can be found in $\mathcal{O}(m + n\log(n))$, if the embedding is given. If the nodes are already sorted by their angle, the separator can be found in $\mathcal{O}(m + n)$.*

Proof. We start by giving another proof for the existence of a $\frac{1}{2}$-balanced of cardinality $\mathcal{O}(\Delta)$. From the proof, the algorithm follows immediately. For a given embedding of G, we select every $\frac{n}{4}th$ node by its angle. Thus, we get 4 nodes x_0, x_1, x_2, x_3 such that in the interior cone of x_i and x_{i+1} with $i \in \{0, 1, 2\}$ lie $\frac{n}{4}$ of the nodes. To do so, we choose x_0 as the node with angle closest to $\frac{3\pi}{2}$. The nodes x_1, x_2 and x_3 follow in clockwise order, like displayed in Fig. 6.4.

In the proof, we will add nodes to a set denoted by S. In the end, the union of neighborhoods of nodes in S is a $\frac{1}{2}$-balanced separator. The set S is initially the empty set. As we saw in Corollary 6.3, the neighborhood of the endpoints of an edge is a separator. We want to separate G in at least two components with $\frac{n}{2}$ nodes or less. To that end, we try to find edges, that have as many internal nodes as possible but less than or equal to $\frac{n}{2}$. This can be done the following way. We start by looking at the cone of x_1 and x_3 that contains x_0. In this cone, we consider edges that have x_0 in one cone and x_1, x_3 (and therefore also x_2) in the other cone. If such an edge exists, let $\{u, v\}$ be the edge that has the most nodes in the cone that x_0 is in. Observe that there are not more than $\frac{n}{2}$ nodes in the cone that x_0 is in due to the choice of x_1 and x_3 having $\frac{n}{2}$ interior and exterior nodes. According to Corollary 6.3, the neighborhood $N(u) \cup N(v)$ separates x_0 from x_1, x_3. If no such edge exist, we add x_0 to S, otherwise we add u and v to S. We do the same for all other nodes x_1, x_2, x_3, each time either adding the considered node x_i with $1 \leq i \leq 3$ or two adjacent nodes to S, that separate x_i from the others and have as many nodes as possible but not more than $\frac{n}{2}$ nodes in the cone that contains x_i.

It is left to prove, that this construction separates G, such that there are no connected components with more than $\frac{n}{2}$ nodes left in $G - N(S)$. Without loss of generality, we assume that the angles between x_i and x_{i+1} with $i \in \{0, 1, 2\}$ are smaller than π. This makes it possible to talk about interior and exterior cones but does not change the arguments of the proof.

We start by looking at the components, the nodes x_i with $0 \leq i \leq 3$ are in. For x_0, there are two cases: Case one, x_0 is in S. If x_0 is in S, there is no edge $\{x, y\}$ from a node x in the interior cone of x_0, x_1 to the node y in the interior cone of x_0, x_3, since these nodes x, y would have been added to S instead of x_0.

Case two, $x_0 \notin S$ then it is in a component with not more than $\frac{n}{2}$ and there are two nodes u, v in S. There is no edge $\{x, y\}$ from a node x in the interior cone of x_0, x_1 to the node y in the interior cone of x_0, x_3, such that both are in the exterior cone of u, v. This is the case, because this edge $\{x, y\}$ would have been chosen to separate x_0 from x_1, x_3 otherwise, since its interior cone contains more nodes than the interior cone of u, v and still less nodes than $\frac{n}{2}$, since it is still between x_1 and x_3. Since this holds for all other nodes x_i with $1 \leq i \leq 3$ as well, no neighboring interior cones, i.e., the cones $x_i, x_{i+1 \bmod 4}$ and $x_i, x_{i-1 \bmod 4}$ with $0 \leq i \leq 3$ have no edges left between each other. Only diagonally, i.e., between $x_i, x_{i+1 \bmod 4}$ and $x_{i+2 \bmod 4}, x_{i-1 \bmod 4}$ with $0 \leq i \leq 1$ there could still be edges. But these diagonal cones all have size not greater than $\frac{n}{4}$, due to the choice of the x_i, thus if two diagonal cones are connected, their connected component still has no more than $\frac{n}{2}$ nodes.

To algorithmically find the nodes in the set S, we look at the nodes sorted by their angle. If we still need to sort them, it takes $\mathcal{O}(n \log n)$. The nodes $x_i, 0 \leq i \leq 3$ can be found by iterating over the nodes and picking every $\frac{n}{4}$th node. During the iteration, we store the index of each node together with the node. Now, to find the edge $\{u, v\}$, for which the cone containing x_i has the greatest cardinality not greater than $\frac{n}{2}$ we proceed as follows. From each x_i we over the nodes in clockwise order until reaching $x_{i+1 \bmod 4}$. When we come to a node with index j, we are interested in the nodes with index k it is adjacent to that are embedded in the cone of x_i and x_{i-1}. If a node with index k is in this cone can be found out by comparing k to the indices of x_i and $x_{i-1 \bmod 4}$. We keep track of the indices j, k, such that $|j - k| \leq \frac{n}{2}$ is maximal among the nodes we have considered so far, i.e., adjacent nodes in the cone of $x_{i+1 \bmod 4}$ and $x_{i-1 \bmod 4}$, x_i is in. If no such pair of indices exists, we add x_i to S, otherwise the nodes with indices j, k are added after having iterated over the whole cone. The correctness of the algorithm follows from the proof above. The run time is obvious as well since we look at the neighborhood of each node only once.

The construction is illustrated in Fig. 6.4 on the right. The cones of x_i and x_{i+1}, $i \in \{0, 1, 2\}$ always contain 3 additional nodes. Observe that x_0 and x_3 are not removed, instead the nodes marked with the orange cones are removed. Although, x_1 is crossed by $\{u_1, u_2\}$, it is removed since u_2 is not in the cone of x_0, x_1. □

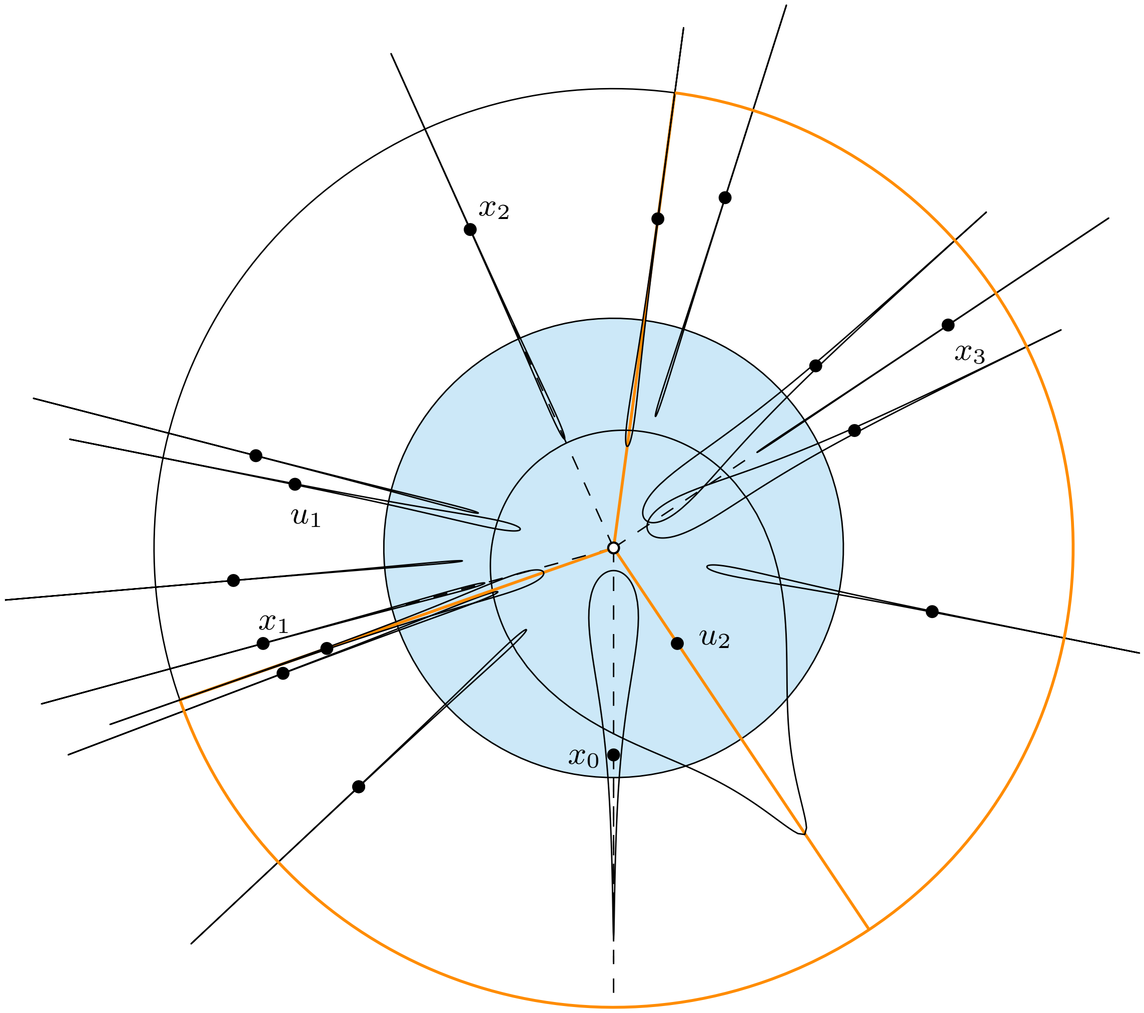

Figure 6.4.: The construction from Theorem 6.7. Starting from x_0 in clockwise direction, after every $\frac{n}{4}$ nodes the next out of x_1 to x_3 follows. To separate the graph, the nodes x_1, x_2 and the nodes that form the orange cones are removed together with their respective neighborhood.

Balanced separators and treewidth are closely related. To get from balanced separators to treewidth we will first recall how separators and separator hierarchies are related. Then, we show how we get from a separator hierarchy to a tree decomposition.

At the beginning of the chapter, separator hierarchies were defined to be a tree of separators in a graph G, such that for a separator S_i that separates some subset of the nodes into two components $G[V_{i_1}], G[V_{i_2}]$, the two children of this separator S_i separate the two components $G[V_{i_1}]$ and $G[V_{i_2}]$, respectively.

Lemma 6.8. *Let G be an SHDG. A $\frac{1}{2}$-balanced separator hierarchy with separators of cardinality $\mathcal{O}(\Delta)$ can be found in $\mathcal{O}(m\log(n) + n\log(n)^2)$ if the embedding is given or $\mathcal{O}(m^2)$ if it is not given.*

Proof. In Theorems 6.6 and 6.7 we showed how to find $\frac{1}{2}$-balanced separators for SHDGs. Since each induced subgraph of an SHDG is an SHDG as well, after separating G into the two components $G[V_{i_1}]$ and $G[V_{i_2}]$ we can apply the algorithm on each component recursively to get a separator hierarchy. Since we always get $\frac{1}{2}$-balanced separators, i.e., the number of nodes is divided by at least 2 in each step, we only need to run the algorithms $\log(n)$ times to get a separator hierarchy.

When the geomerty is not given, this still runs in $\mathcal{O}(m^2)$ with Algorithm 6.1, which can be seen with the third case of the master theorem. The run time with Theorem 6.7 goes up to $\mathcal{O}(m\log(n) + n\log(n)^2)$. □

This means we have an efficient way to get separator hierarchies. The following Lemma 6.9 bridges the gap between separator hierarchies and tree decompositions. Recall that in a separator hierarchy (T, S), a separator $S_i \in S$ is separated from all other separators in S except for its ancestors and descendants in T:

Lemma 6.9 (Lemma 8 in [BFK])**.** *Let (T, S) be a separator hierarchy of G. For each node i of T, let X_i be the union of S_i and all separators S_j for which j is an ancestor of i in T. Then $(T, X = x_0, ..., X_t)$ is a tree decomposition of G.*

Now we have all the tools to construct a tree decompositon and can conclude this section with Theorem 6.10 on the treewidth of SHDGs.

Theorem 6.10. *The treewidth of SHDGs is in $\mathcal{O}(\Delta\log(n))$. The associated tree decomposition can be found in $\mathcal{O}(m\log(n) + n\log(n)^2)$ if the embedding is given or $\mathcal{O}(m^2)$ if it is not given.*

Proof. Since all separators in the separator hierarchy (T, S) of G from Lemma 6.8 were of cardinality $\mathcal{O}(\Delta)$ and T has height $\log(n)$, Lemma 6.9 yields a tree decomposition with bags of size $\mathcal{O}(\Delta\log(n))$ and thus the treewidth is in $\mathcal{O}(\Delta\log(n))$.

The run times for the tree decomposition follow directly from the run times to get the separator hierarchies in Lemma 6.8 with only linear overhead from 6.9 by top-down forming the union of the separators in S along T. □

6.2. Grids in SHDGs

In this section we study what grids in SHDGs can look like. This is related to treewidth in that there exists a function $g(t) \in \mathcal{O}(t^{98+o(1)})$ such that every graph with treewidth larger than $g(t)$ contains a $t \times t$ grid as a minor [CC16]. This bound might be rather high, but if we can show that an SHDG that contains a grid of size t has exponentially many nodes in the grid size i.e., that the grid size is in $\mathcal{O}(\log n)$, this still means that the treewidth is in $\mathcal{O}(\log n)$ as well. In this section, we formulate multiple conjectures on what grids in SHDGs look like. We begin by discussing why we think that for every grid size t there is an SHDG G that has a constant clique number. The construction in this discussion is almost proof-like because we believe it to be helpful in possible proofs of the following conjectures as well. Next up, we discuss why we believe that an SHDG with constant clique number that contains a grid of size t has $\Omega(c^t)$ nodes for a constant $c > 1$. This discussion will be on a higher level of abstraction.

To construct an SHDG with constant clique number that contains a grid of arbitrary size t, we first need a few definitions. The *distance of a point p to a line l* is the minimum distance among all points on l to p. Let l be a line in the hyperbolic pane. The set of points with distance r from l is called the *hypercircle H* with *axis l* and *radius r*. We call the points with distance smaller r from l the *interior points* of H. For a hypercircle H with axis l we call the points of H on one side of l a *half-circle* of H.

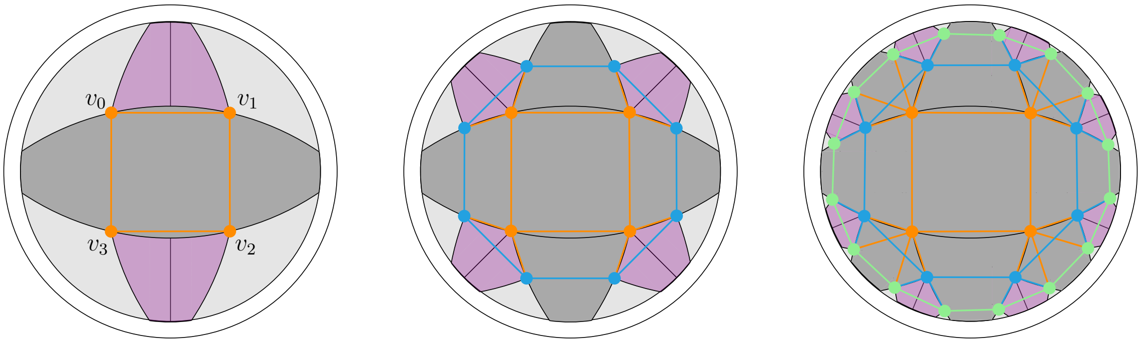

Figure 6.5.: First 3 iterations of the construction in Conjecture 6.11. Nodes are placed in the intersections of the purple hypercircles with previously placed hypercircles that are not in the interior of any previously placed hypercircle.

Conjecture 6.11. *For any grid size t, there is an SHDG G with constant clique number ω that has a $t \times t$ grid as a minor.*

Discussion: We will use hypercircles and place nodes in their intersections. This way, we can ensure that certain nodes have a distance larger than R from another. We first describe the construction, then we discuss why the clique number is constant and finish by arguing why we think that the construction works for grids of arbitrary size.

If we set the radius of a hypercircle H with axis l to $\frac{R}{2}$, we know that for a point $p \in H$ on one half-circle of H there is exactly one point $q \in H$ on the other half-circle that has distance R from p. More precisely, p and q are adjacent if and only if p mirrored at l is q. We further know, that every point except q that is not on the same side of l as p and that is not in the interior of H has distance greater R from p.

In the following construction, we will always put nodes symmetrically to both sides of hypercircles, these nodes are therefore adjacent.

The construction. Let H_i denote a hypercircle with radius $\frac{R}{2}$ and axis l_i going through the origin for $i \in \mathbb{N}_0$. In this discussion, we will specify the angles of the different axes with respect to the axis l_0. We begin by choosing l_0 to be a horizontal line through the origin. We construct the hypercircles and place the nodes in iterations: in iteration 1, we set axis l_1 orthogonal to l_0 through the origin. The hypercircles H_0 and H_1 are depicted in Fig. 6.5 on the left. We now place a node in each of the 4 intersections of H_0 and H_1. We refer to the nodes by v_0 to v_3, where v_0 has angle $\frac{3\pi}{4}$ and the other nodes are numbered in clockwise order. As discussed above they form a C_4, where v_i and $v_{i+1 \bmod 4}$ for $0 \leq i \leq 3$ are adjacent.

While the axis l_1 had angle $\frac{\pi}{2}$ with respect to l_0, the axes of the next iteration will have angle $\frac{\pi}{4}$ and angle $\frac{3\pi}{4}$, as depicted in Fig. 6.5 in the middle. This way, they go through 2 nodes placed in the previous iteration. We place nodes in the intersection points, that are not in the interior of any previously placed hypercircles. Due to symmetry, they form a cycle again.

We continue placing hypercircles until the next iteration of hypercircles placed would intersect the boundary of the ground space in the interior of or on another hypercircle that was already placed. The angles for the axes are $\frac{k\pi}{2^i}$ where i is the number of the iteration starting at 1 with l_0 already placed and $k \in 1, ..., 2^i$ with odd k. As described above, we place nodes in the intersection points with previously placed hypercircles, that are not in the interior of any previously placed hypercircles. Due to symmetry, they form cycles every time.

A grid with constant clique number. So far we have only shown that the construction yields cycles with an exponentially growing number of nodes in each iteration. We look at which of the nodes are adjacent now. Fig. 6.6 shows the part of the graph that is relevant for the discussion in a black box. Recall that the axes of an iteration go through 2 nodes of the previous iteration if there are any such nodes. Let v_1 be one of these nodes and l_i be the axis through v_1. We denote the two nodes that are placed next to v_1 by v_2 and v_3. According to Lemma 2.1, v_1 is adjacent to both of them, since they are adjacent and v_1 is between them and closer to the origin. All other nodes placed in the iteration in which v_2 and v_3 were placed are on the other side (where the side is defined by the axis) of a whole hypercircle and therefore not adjacent to v_1. Thus, v_1 is only adjacent to v_2 and v_3 from this iteration. We start another iteration. Since the nodes of this iteration form a cycle again, v_1 is adjacent to at least two of them according to Lemma 2.1. From the construction we know that it has to be the two nodes at intersections with the hypercircle v_1 is on as well. We denote these nodes by v_4 and v_5. This time, it is not obvious that v_1 is not adjacent to any other nodes that were added in this iteration since there are two more nodes that are not on the other side of a whole hypercircle. We denote these nodes by v_6 and v_7. In fact, it is possible, that both of them are adjacent to v_1. The points v_6 and v_7 are on half-circles together with v_1 which we denote by A_{v_6} and A_{v_7}. Instead of placing the nodes v_6 and v_7 in this iteration, we continue with the next iteration, until the next node on the half-circles A_{v_6} and A_{v_7} would be placed with distance larger than R from v_1.

We skip placing nodes, every time a node is adjacent to a node from a previous iteration for another reason than Lemma 2.1. This leaves us with nodes only adjacent to two nodes from each following iteration. The construction after 4 iterations in which nodes were placed is illustrated in Fig. 6.7. Observe that edges from nodes in the blue and orange iteration have no common neighbor in the black iteration. Since this is the case for any 4 consecutive iterations, the clique number is constant.

There are enough levels. If we want to construct graphs that contain a grid of size t, we need to have $\mathcal{O}(t)$ iterations in which nodes are placed. The construction terminates before having done enough iterations if the next iteration of hypercircles placed would

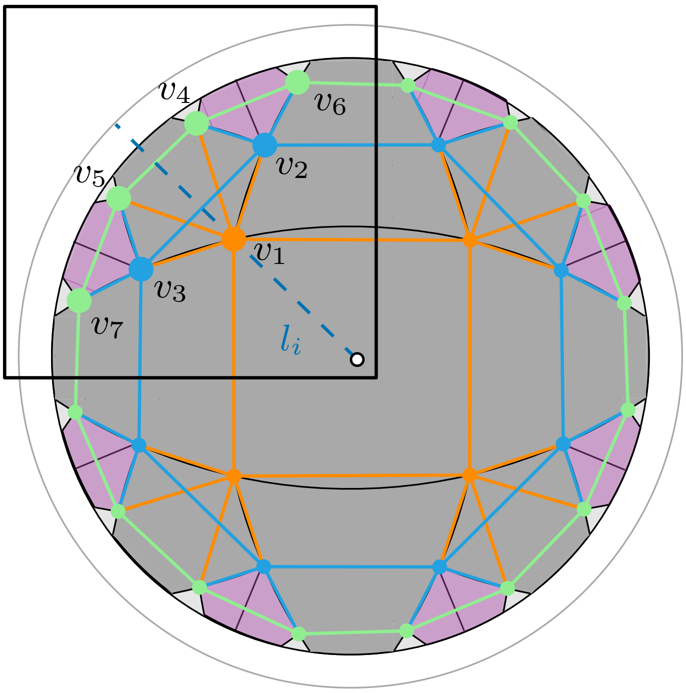

Figure 6.6.: Possibly adjacent nodes to v_1 in the construction of Conjecture 6.11. All but v_6 and v_7 are adjacent to v_1 by Lemma 2.1. The construction ensures that v_6 and v_7 are not adjacent to v_1.

intersect the boundary of the ground space in the interior or on another hypercircle that was already placed. The intersections of each hypercircle with the boundary of the ground space can be interpreted as corners of an equilateral triangle with its third corner at the origin. For these triangles, we showed that the angle at the origin gets arbitrarily small in Lemma 2.4. Thus, we can choose R big enough, such that any number of hypercircles can be placed.

This is only a conjecture since it is not enough to show that we can place any number of hypercircles. In the construction, we sometimes skip placing nodes until there is an intersection of hypercircles with distance greater R to an already placed node. It is necessary to show that such an intersection point always exists in our construction. In Appendix A, we discuss why we think that there can be any number of points on a half-circle that have pairwise distance R or more but this only considers points on one half-circle which is not always the case in our construction. △

Conjecture 6.12. *Let G be an SHDG with constant clique number ω that contains a grid of size t as a minor. Then, G has $\Omega(c^t)$ nodes for $c > 1$.*

Discussion: Every grid of size t contains $\lfloor \frac{t}{2} \rfloor$ cycles like depicted in Fig. 6.9 on the left, where the considered cycles are depicted in full lines while the other edges are depicted in dotted lines. To be able to refer to these cycles, we say *cycle of level one* for the cycle of 8 nodes in the middle, *cycle of level two* for the cycle of 16 nodes around the cycle of level one and so on. In this discussion, we are mainly interested in these cycles. If G contains a grid as a minor, the cycles present in the grid were present in G already before contracting edges or removing anything – possibly with more nodes but a cycle nevertheless. Therefore, we argue on where the cycles of a $t \times t$ grid are embedded as representatives of the cycles of G that we are interested in.

Recalling Lemma 2.1, if a node with smaller radius is between two adjacent nodes with larger radius, they are all adjacent. In this case, we say the node with smaller radius is

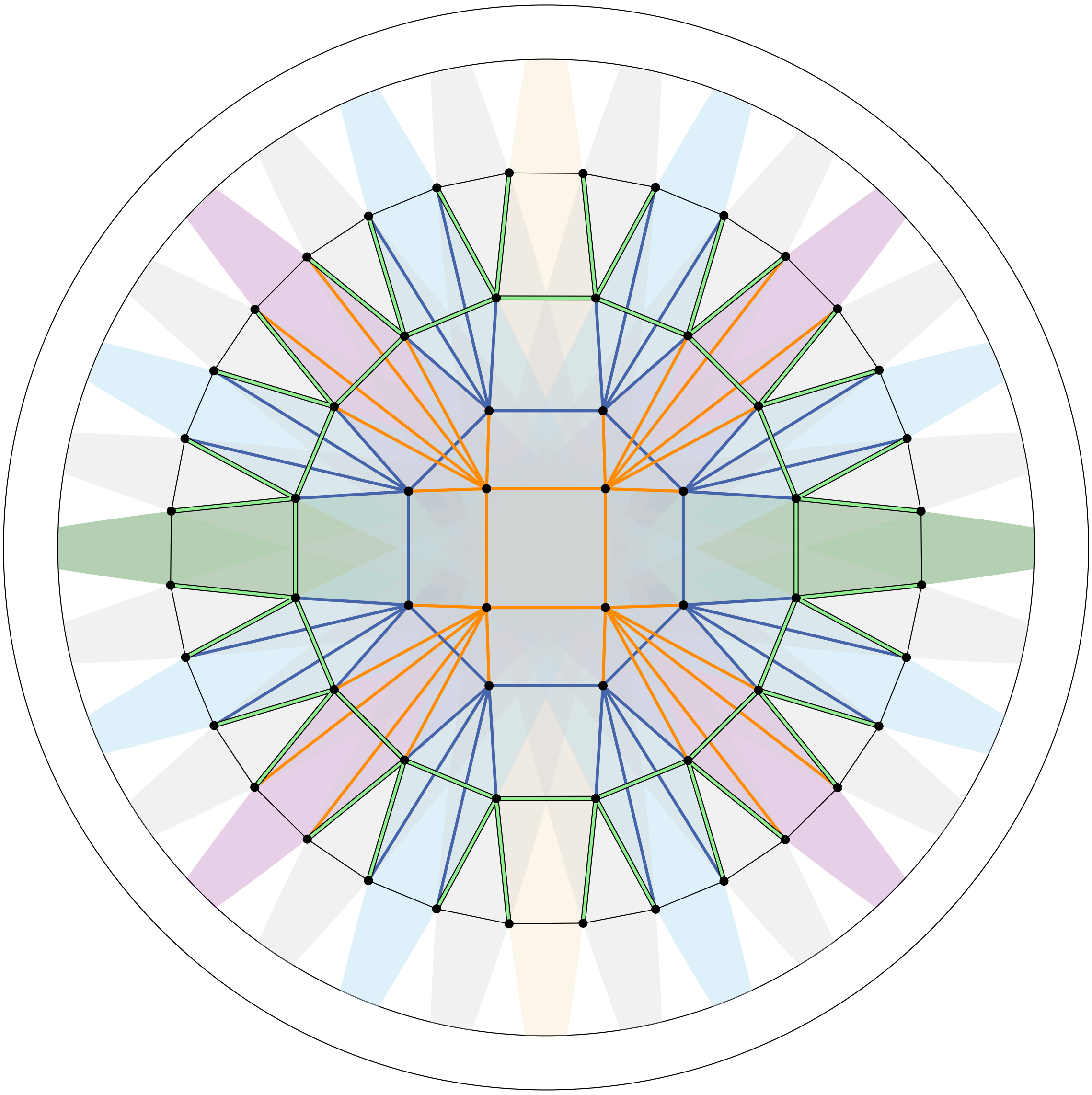

Figure 6.7.: The construction from Conjecture 6.11 for 4 iteration in which nodes were placed.

above the edge between the other two nodes. We emphasize once more, that the edges depicted in Fig. 6.7 are only edges of the cycles or edges implied by Lemma 2.1. We focus on the part of the graph from Fig. 6.7 depicted in Fig. 6.8. We refer to the nodes of each iteration by the color of their respective cycle. The reason why we think that there are exponentially many nodes is: the nodes from the early iterations, i.e., the orange and blue nodes, are adjacent to at least two nodes from each of the later iterations (the endpoints of the edge they are above of). For the clique number to stay 3, we therefore need 3 edges, i.e., 4 nodes, in the green iteration. For the next iteration, i.e., the black nodes, we observe it is the same scenario we just had with orange and blue nodes, this time for each of the blue nodes together with 2 of the green nodes – again, there need to be 4 black nodes for each blue node. The orange node must also be above an edge that no other node is above of for the clique number to stay 3.

If the clique number is bigger, we can allow for more nodes above edges of the next levels. But once there are enough nodes on lower levels, we still need an edge for every $\omega - 2$ nodes.

Therefore, if the $\lfloor \frac{t}{2} \rfloor$ cycles of the grid are embedded like schematically depicted in Fig. 6.9 in the middle, with the origin in the interior of the curve formed by their edges, we need $\Omega(c^t)$ nodes for a $c > 1$, to have constant clique number. It is possible that c is always 2 but we cannot answer this with certainty.

If the cycles do not contain the origin, like schematically depicted in Fig. 6.9 on the right, the equivalent to a cycle of level i is a *path of level* i, depicted by the colored paths. Since the paths are part of the cycles of the grid, the path nodes are above some of the edges of the cycles they belong to. This only leads to more nodes needed for the clique number to stay the same. Thus, there have to be exponentially many nodes in this case as well. △

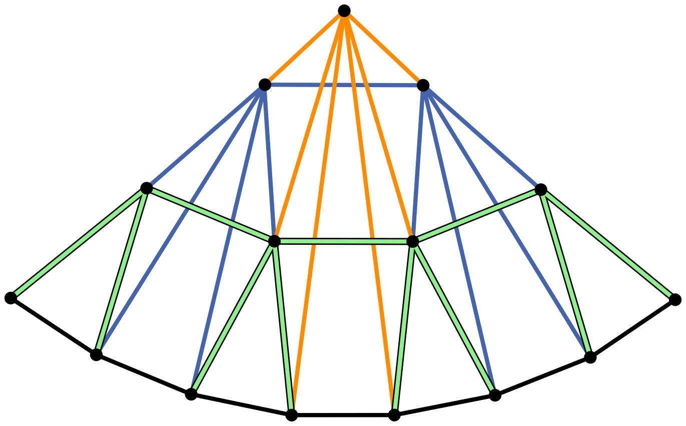

Figure 6.8.: A part of the grid from Fig. 6.7 that illustrates why Lemma 2.1 already requires exponentially many nodes.

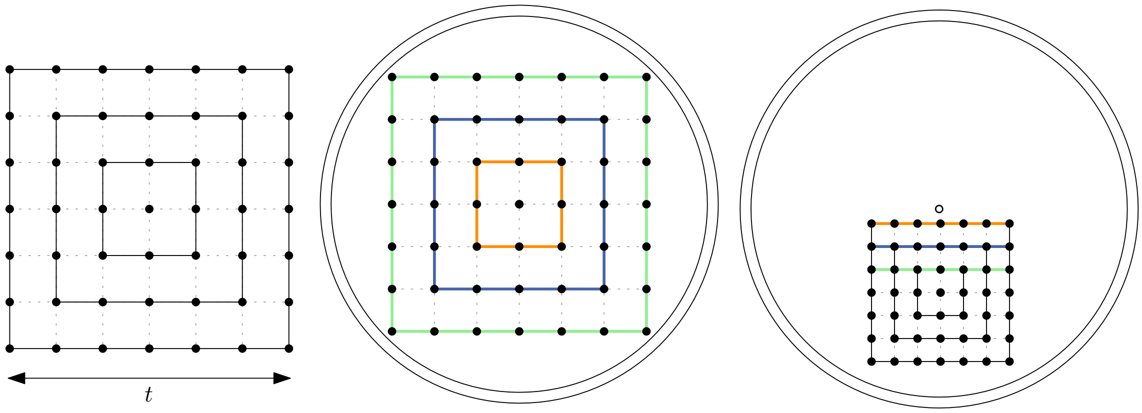

Figure 6.9.: On the left: a $t \times t$ grid always contains $\lfloor \frac{t}{2} \rfloor$ cycles. In the middle: if the cycles of the grid are embedded with the origin in the interior of the curves formed by the cycle edges, the number of nodes has to be exponential for the clique number to be constant like in the construction of Conjecture 6.11. On the right: if the cycles do not have the origin in the interior of their curves, the colored paths enforce even more nodes to keep the clique number constant.

7. Conclusion

In Chapter 3, we showed that Euclidean unit disk graphs (EDGs) are a subclass of HDGs but incomparable with SHDGs. We further showed that HDGs and SHDGs are incomparable to chordal graphs. For circular arc graphs (CAGs) we showed that not even the inner neighborhood of an SHDG is a CAG but that every unit CAG is an SHDG and thus an HDG, which implies that unit interval graphs and proper interval graphs are SHDGs.

In Chapter 4, we showed that recognizing HDGs is NP-hard and a member of $\exists\mathbb{R}$.

In Chapter 5, we showed that maximum cliques of all graphs that admit a CNEO1 can be found in $\mathcal{O}(\sum_{v\in V} |N(v)|^3)$ and that SHDGs admit a CNEO1. We further showed that HDGs admit a CNEO2 for which an efficient algorithm exists to find all maximum cliques. We found a gap in the proof of correctness of this algorithm though which we were not able to fix.

In Chapter 6, we showed that the treewidth of an SHDG is limited by $4\Delta\log(n)$ by contributing algorithms that find $\frac{1}{2}$-balanced separators, $\frac{1}{2}$-balanced separator hierarchies and tree decompositions in $\mathcal{O}(m^2)$ without having an embedding given. If an embedding is given, we improved this run time even further. Concerning grids, we conjectured that while keeping the clique number constant, an SHDG can have a grid minor of any size. This comes at the cost of also having exponentially many nodes though.

For related graph classes, it is interesting to further investigate the relation of SHDGs to CAGs and interval graphs. We think the most promising relation to further investigate is the relation between interval graphs and SHDGs. This might not only lead to a result on the relation of the two classes but also be helpful when considering CAGs. Our idea is based on proper interval graphs being unit interval graphs for which we showed how to find an SHDG embedding. To find an SHDG embedding for a general interval graph G with a given interval model, it might be possible to first embed the induced subgraph G' for which the interval model consists of all intervals that are not fully contained in another interval, i.e., a subgraph that is a proper interval graph. Thus, there is a unit interval representation of G' for which we know how to embed it. For the graph that is left after removing G', we again look for the induced proper interval graph as described above and embed it with respect to the already embedded part of G.

Concerning recognition, there are open questions for both HDGs and SHDGs. The NP-hardness of recognizing HDGs paired with membership in $\exists\mathbb{R}$ poses the question of whether recognizing HDGs is $\exists\mathbb{R}$-complete, as it is in the case of EDGs. It might be helpful to limit the range from which the threshold distance R can be chosen to construct gadgets for a possible reduction. To that end, one could use a combination of star graphs and cages, like they were used to show NP-hardness. While star graphs limit R from below, cages limit R from above. For SHDGs, we do not know anything about the complexity of recognition and only little about what properties of the graph we might want to use to efficiently recognize them or to show NP-hardness. In this thesis, we only showed that the existence of a cobipartite elimination scheme by itself is insufficient for a graph to be an SHDG. The limited ground space of SHDGs leads to *a lot of structure* in SHDGs though. Therefore, we think efficient recognition might be possible.

For maximum cliques, it is unclear if the algorithm by Raghavan *et al.* [RS03] is correct. Fixing the gap in their proof would show that maximum cliques can be found efficiently for both EDGs and HDGs. For SHDGs, the run time of our algorithm ($\mathcal{O}(\sum_{v \in V} |N(v)|^3)$) to find maximum cliques is dominated by checking if a graph becomes bipartite when a node is removed. If there is a way to speed up consecutive checks if a graph is bipartite when removing nodes, the run time can be improved to $\mathcal{O}(\sum_{v \in V} |N(v)|^{2.5})$ at which point finding a maximum matching in a bipartite graph dominates the run time. One idea to do so is to keep track of the induced odd cycles that are necessary for the graph to be not bipartite. They can be found via BFS where each odd cycle leads to a (not necessarily distinct) cross edge. The main task here is to avoid having to run a BFS every time to find a new cycle.

For separators and treewidth, our algorithm to find balanced separators with run time $\mathcal{O}(m^2)$ currently checks all edges, if their endpoints separate the graph in the most balanced way. By only checking edges for which the endpoints lead to a more balanced separator, this run time could be improved. This might work by only considering the neighborhoods of the endpoints of previously checked edges that made the size of the separator plus the size of the largest connected component smaller. This is promising since for a fixed separator S checking edges that have both endpoints in the smaller connected component will never make the bigger connected component any smaller. Furthermore, when only checking edges that have endpoints in the neighborhoods of previously checked edges, it might be possible to check for the largest connected component faster than running a DFS every time. We see the most potential in grids, though. The conjectures we presented give a promising direction for further study of grids in SHDGs and we even believe that the number of nodes is exponential in the grid size, even when the clique number is not constant.

Bibliography

[ABF15] Mohammed Amin Abdullah, Michel Bode, and Nikolaos Fountoulakis. Typical distances in a geometric model for complex networks. *arXiv preprint arXiv:1506.07811*, 2015.

[AZ18] Aistis Atminas and Viktor Zamaraev. On forbidden induced subgraphs for unit disk graphs. *Discrete & Computational Geometry*, 60(1):57–97, 2018.

[BFK] Thomas Bläsius, Tobias Friedrich, and Anton Krohmer. Hyperbolic random graphs: Separators and treewidth.

[BFK18] Thomas Bläsius, Tobias Friedrich, and Anton Krohmer. Cliques in hyperbolic random graphs. *Algorithmica*, 80(8):2324–2344, 2018.

[BFKS21] Thomas Bläsius, Tobias Friedrich, Maximilian Katzmann, and Daniel Stephan. Routing in strongly hyperbolic unit disk graphs. *arXiv preprint arXiv:2107.05518*, 2021.

[BK98] Heinz Breu and David G Kirkpatrick. Unit disk graph recognition is np-hard. *Computational Geometry*, 9(1-2):3–24, 1998.

[BP93] Jean RS Blair and Barry Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, pages 1–29. Springer, 1993.

[BRMW13] Ulrik Brandes, Garry Robins, Ann McCranie, and Stanley Wasserman. What is network science? *Network science*, 1(1):1–15, 2013.

[CC16] Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *Journal of the ACM (JACM)*, 63(5):1–65, 2016.

[CCJ90] Brent N Clark, Charles J Colbourn, and David S Johnson. Unit disk graphs. *Discrete mathematics*, 86(1-3):165–177, 1990.

[CCPS09] William J Cook, William Cunningham, William Pulleyblank, and A Schrijver. Combinatorial optimization. *Oberwolfach Reports*, 5(4):2875–2942, 2009.

[DGS14] Guillermo Durán, Luciano N Grippo, and Martín D Safe. Structural results on circular-arc graphs and circle graphs: a survey and the main open problems. *Discrete Applied Mathematics*, 164:427–443, 2014.

[FK18] Tobias Friedrich and Anton Krohmer. On the diameter of hyperbolic random graphs. *SIAM Journal on Discrete Mathematics*, 32(2):1314–1334, 2018.

[Gar07] Frédéric Gardi. The roberts characterization of proper and unit interval graphs. *Discrete Mathematics*, 307(22):2906–2908, 2007.

[Gav74] Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47–56, 1974.

[GPP12] Luca Gugelmann, Konstantinos Panagiotou, and Ueli Peter. Random hyperbolic graphs: degree sequence and clustering. In *International Colloquium on Automata, Languages, and Programming*, pages 573–585. Springer, 2012.

[HK73] John E Hopcroft and Richard M Karp. An n^5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.

[HP] Sariel Har-Peled. On complexity, sampling, and ϵ-nets and ϵ-samples. `https://sarielhp.org/teach/13/b_574_rand_alg/lec/20_vc_dim.pdf`, Accessed: 2022-1-3.

[Inc] Wolfram Research, Inc. Mathematica, Version 12.1. Champaign, IL, 2020.

[KB20] Sándor Kisfaludi-Bak. Hyperbolic intersection graphs and (quasi)-polynomial time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1621–1638. SIAM, 2020.

[Kle07] Robert Kleinberg. Geographic routing using hyperbolic space. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 1902–1909. IEEE, 2007.

[KLS$^+$19] István A Kovács, Katja Luck, Kerstin Spirohn, Yang Wang, Carl Pollis, Sadie Schlabach, Wenting Bian, Dae-Kyum Kim, Nishka Kishore, Tong Hao, et al. Network-based prediction of protein interactions. *Nature communications*, 10(1):1–8, 2019.

[KM12] Ross J Kang and Tobias Müller. Sphere and dot product representations of graphs. *Discrete & Computational Geometry*, 47(3):548–568, 2012.

[KPK$^+$10] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[Kro16] Anton Krohmer. *Structures & algorithms in hyperbolic random graphs*. PhD thesis, Universität Potsdam, 2016.

[PKBV10] Fragkiskos Papadopoulos, Dmitri Krioukov, Marián Boguná, and Amin Vahdat. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, 2010.

[Rob69] Fred S Roberts. Indifference graphs. *Proof techniques in graph theory*, pages 139–146, 1969.

[RS03] Vijay Raghavan and Jeremy Spinrad. Robust algorithms for restricted domains. *Journal of algorithms*, 48(1):160–172, 2003.

[StE] Question on math stackexchange. `https://math.stackexchange.com/questions/4379358/how-to-show-that-frac-sinhb-sinha-leq-fracba-for-a-geq-b0`, Accessed: 2022-2-11.

[Tuc70] Alan Tucker. Characterizing circular-arc graphs. *Bulletin of the American Mathematical Society*, 76(6):1257–1260, 1970.

Appendix

A. Points on Half-circles of Hypercircles

Conjecture 7.1. *Let H be a hypercircle with radius $\frac{R}{2}$ and axis l. Furthermore, let p, q be two points on one of the half-circles of H that have distance R and let p' and q' be the points on l that have orthogonal distance $\frac{R}{2}$ to p and q, respectively. Then, the distance between p' and q' converges to $|\log(\frac{1}{2})|$.*

Discussion: We want to discuss what the distance between p' and q' is. To that end, we have a look at Fig. A.1, where the segment of l between p' and q' and the segment of the hypercircle between p and q is shown. We want to find the length of c, i.e., the length of the geodesic between p and q with respect to x. This way, we can find out for what value of x, p and q have distance $c = R$. From hyperbolic tangent formula $\tan(\alpha) = \frac{\tanh(\frac{R}{2})}{\sinh(x)}$, we know

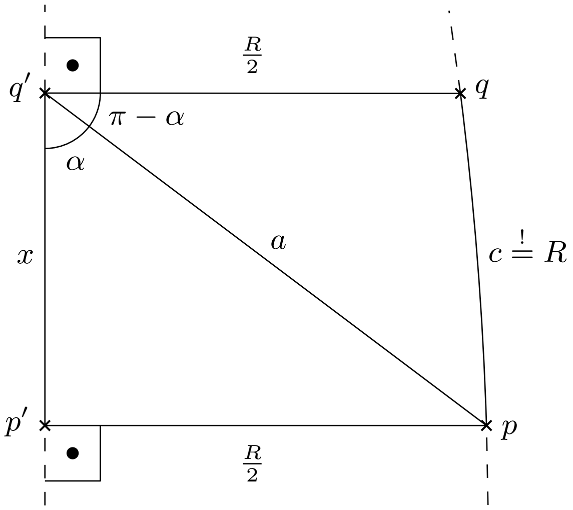

Figure A.1.:

that $\alpha = \operatorname{arctan}(\frac{\tanh(\frac{R}{2})}{\sinh(x)})$. From hyperbolic sine formula $\sin(\alpha) = \frac{\sinh(\frac{R}{2})}{\sinh(a)}$, we know that $a = \operatorname{arsinh}(\frac{\sinh(\frac{R}{2})}{\sin(\alpha)})$. Since there is a right angle at q', the angle between a and b in Fig. A.1 is $\pi - \alpha$. From hyperbolic cosine law $\sinh(a)\sinh(\frac{R}{2})\cos(\pi - \alpha) = \cosh(a)\cosh(\frac{R}{2}) - \cosh(c)$

and Mathematica [Inc] we get
$c = \operatorname{arcosh}\left(\cosh\left(\frac{R}{2}\right)\sqrt{\cosh^2\left(\frac{R}{2}\right)\sinh^2(x)+\sinh^2\left(\frac{R}{2}\right)+1}+\sinh\left(\frac{R}{2}\right)\cosh\left(\frac{R}{2}\right)\sinh(x)\right).$
For real numbers, this simplifys to $c = \operatorname{arcosh}(\frac{1}{2}(\cosh(R+x)+\cosh(x)))$. We want to know for what value of x the formulas above yield $c = R$, i.e., how long the segment between p' and q' is, given that p and q have distance R from another. Using Mathematica again, we numerically observed that x converges to $|log(\frac{1}{2})| \approx 0.69$ for increasing R. △

In the next conjecture, we will need that we can place a lot of points on one half-circle such that they at least have a distance of R and radius not larger than R.

Conjecture 7.2. *Let H be a hypercircle with radius $\frac{R}{2}$ and axis l. For increasing R, the number of points we can place on one half-circle of H such that they have pairwise distance at least R and radius not greater than R goes to infinity for R to infinity.*

Discussion: We consider points on the axis with pairwise distance at least $|log(\frac{1}{2})| + \varepsilon$ for $\varepsilon > 0$. Assuming Conjecture 7.1 holds, for R large enough the associated points on one half-circle of H have distance at least R. Since the number of points on l, i.e., $\left\lfloor \frac{R}{|log(\frac{1}{2})|+\varepsilon} \right\rfloor$, goes to infinity for R to infinity, the number of associated points on the the half-circle goes to infinity as well. This is only a conjecture since it is not sufficient for only the points on the axis to have a radius not larger than R but this needs to hold for the points on the half-circle as well. △