About the Analysis of Algorithms on Networks with Underlying Hyperbolic Geometry

Maximilian Katzmann

Universitätsdissertation
zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

in der Wissenschaftsdisziplin
Theoretische Informatik

eingereicht an der
Digital-Engineering-Fakultät
der Universität Potsdam

Datum der Disputation: 20. Januar 2023

Betreuer

Prof. Dr. Tobias Friedrich
Hasso Plattner Institute, University of Potsdam

Gutachter

Prof. Dr. Marcos Kiwi
Universidad de Chile

Prof. Dr. Tobias Müller
University of Groningen

Prof. Dr. Sebastian Siebertz
Universität Bremen

Published online at the
Institutional Repository of the University of Potsdam:
https://doi.org/10.25932/publishup-58296
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-582965

Abstract

Many complex systems that we encounter in the world can be formalized using *networks*. Consequently, they have been in the focus of computer science for decades, where algorithms are developed to understand and utilize these systems.

Surprisingly, our theoretical understanding of these algorithms and their behavior in practice often diverge significantly. In fact, they tend to perform much better on real-world networks than one would expect when considering the theoretical worst-case bounds. One way of capturing this discrepancy is the *average-case analysis*, where the idea is to acknowledge the differences between practical and worst-case instances by focusing on networks whose properties match those of real graphs. Recent observations indicate that good representations of real-world networks are obtained by assuming that a network has an underlying hyperbolic geometry.

In this thesis, we demonstrate that the connection between networks and hyperbolic space can be utilized as a powerful tool for average-case analysis. To this end, we first introduce *strongly hyperbolic unit disk graphs* and identify the famous *hyperbolic random graph* model as a special case of them. We then consider four problems where recent empirical results highlight a gap between theory and practice and use hyperbolic graph models to explain these phenomena theoretically. First, we develop a *routing scheme*, used to forward information in a network, and analyze its efficiency on strongly hyperbolic unit disk graphs. For the special case of hyperbolic random graphs, our algorithm beats existing performance lower bounds. Afterwards, we use the hyperbolic random graph model to theoretically explain empirical observations about the performance of the *bidirectional breadth-first search*. Finally, we develop algorithms for computing optimal and nearly optimal *vertex covers* (problems known to be NP-hard) and show that, on hyperbolic random graphs, they run in polynomial and quasi-linear time, respectively.

Our theoretical analyses reveal interesting properties of hyperbolic random graphs and our empirical studies present evidence that these properties, as well as our algorithmic improvements translate back into practice.

Zusammenfassung

Viele komplexe Systeme mit denen wir tagtäglich zu tun haben, können mit Hilfe von *Netzwerken* beschrieben werden, welche daher schon jahrzehntelang im Fokus der Informatik stehen. Dort werden Algorithmen entwickelt, um diese Systeme besser verstehen und nutzen zu können.

Überraschenderweise unterscheidet sich unsere theoretische Vorstellung dieser Algorithmen jedoch oft immens von derem praktischen Verhalten. Tatsächlich neigen sie dazu auf echten Netzwerken viel effizienter zu sein, als man im schlimmsten Fall erwarten würde. Eine Möglichkeit diese Diskrepanz zu erfassen ist die *Average-Case Analyse* bei der man die Unterschiede zwischen echten Instanzen und dem schlimmsten Fall ausnutzt, indem ausschließlich Netzwerke betrachtet werden, deren Eigenschaften die von echten Graphen gut abbilden. Jüngste Beobachtungen zeigen, dass gute Abbildungen entstehen, wenn man annimmt, dass einem Netzwerk eine hyperbolische Geometrie zugrunde liegt.

In dieser Arbeit wird demonstriert, dass hyperbolische Netzwerke als mächtiges Werkzeug der Average-Case Analyse dienen können. Dazu werden *stark-hyperbolische Unit-Disk-Graphen* eingeführt und die bekannten *hyperbolischen Zufallsgraphen* als ein Sonderfall dieser identifiziert. Anschließend werden auf diesen Modellen vier Probleme analysiert, um Resultate vorangegangener Experimente theoretisch zu erklären, die eine Diskrepanz zwischen Theorie und Praxis aufzeigten. Zuerst wird ein *Routing Schema* zum Transport von Nachrichten entwickelt und dessen Effizienz auf stark-hyperbolischen Unit-Disk-Graphen untersucht. Allgemeingültige Effizienzschranken können so auf hyperbolischen Zufallsgraphen unterboten werden. Anschließend wird das hyperbolische Zufallsgraphenmodell verwendet, um praktische Beobachtungen der *bidirektionalen Breitensuche* theoretisch zu erklären und es werden Algorithmen entwickelt, um optimale und nahezu optimale *Knotenüberdeckungen* zu berechnen (NP-schwer), deren Laufzeit auf diesen Graphen jeweils polynomiell und quasi-linear ist.

In den Analysen werden neue Eigenschaften von hyperbolischen Zufallsgraphen aufgedeckt und empirisch gezeigt, dass sich diese sowie die algorithmischen Verbesserungen auch auf echten Netzwerken nachweisen lassen.

Acknowledgments

While my journey as a PhD student started a couple of years ago, it is as though it went by in an instant. Yet, so much has happened in this time and I am grateful for all the memories that were made. Of course, none of it would mean a thing, had I not shared them with the friends I had and the ones I made on the way.

First and foremost, I want to thank Tobias Friedrich for almost a decade of continued advice and support, for the challenges you posed and the assistance you gave as I tried mastering them, which eventually helped me to grow as a person. You provided protection from all kinds of uncertainties that arise in the world of academia and allowed me to focus on what is the most fun and rewarding for me: research. Thank you.

Special thanks also go to Thomas Bläsius. What started as a mentoring quickly grew into a friendship that I am very grateful for. You taught me so much about how to be a good scientist and helped me cope with the setbacks and the stress that go along with it. This continues on from mental to physical gymnastics, where your repeated motivations to joining beach volleyball or bouldering guided me to leading a healthier life. Thank you.

My time at the chair was accompanied by so many more people that helped me in one way or the other, including Samuel Baguley, Katrin Casel, Sarel Cohen, Ankit Chauhan, Philipp Fischbeck, Andreas Göbel, Hazar Harmouch, Katrin Heinrich, Timo Kötzing, Martin Krejca, Simon Krogmann, Anton Krohmer, Gregor Lagodzinski, Pascal Lenzner, Xiaoyue Sherry Li, Michael Loster, Nadym Mallek, Anna Melnichenko, Louise Molitor, Stefan Neubert, Aikaterini Niklanovits, Marcus Pappik, Francesco Quinzan, Aishwarya Radhakrishnan, Ralf Rothenberger, Janosch Ruff, Leon Schiller, Martin Schirneck, Karen Seidel, Christopher Weyand, Marcus Wilhelm, Ziena Zeif, and even Vanja Doskoč. Ranging from fun activities like the board games after lunch or sports, to discussing research and tackling the challenges that were thrown at us, you were there with me and I am thankful for that.

And of course, I also received support from outside university. My bouldering friends Hendrik, Madeleine, and Simon presented delightful distractions from

work. My brother Lukas was with me the whole time, always quick to respond for fun exchanges and to discuss all aspects of life. I am also grateful for the support from my parents Sabine and Andreas, my grandparents Hildrun and Volker, as well as the family I found in Janine, Daniel, Lina, Leo, and Jens.

Last, but most certainly not least (as she made sure I included here), I want to thank Denise for everything. You provide me with daily support, bring balance to my life, and help me cope in trying times. Whether it is going for a walk, laughing at my jokes, or listening to my ideas, you make sure that I feel well and improve every aspect of my life. I am looking forward to our journey ahead. Safe travels.

Contents

1 Introduction

Networks are a powerful tool to capture the ubiquitous relationships between entities that are encountered on all scales in our universe. Starting at the atomic level where networks describe the relationships between atoms that connect to form molecules, on to protein-interaction networks describing the ways certain molecules interact, and up to the human connectome capturing the relationships between neurons in our brains. This continues on larger scales, where social networks represent relationships between people, to the internet that connects the whole world, eventually reaching the cosmic web of galaxies. As a general framework encompassing all of the above, networks or *graphs* model entities as *vertices* and the relationships as *edges* between them.

Some networks are small. The water molecule, for example, consists of two hydrogen atoms that connect to a single oxygen atom, i.e, there are two edges among three vertices. Understanding and working with such a network is rather straight-forward. On the other hand, we often encounter huge networks like the internet containing billions of connected devices. Such graphs are harder to process. Even simple tasks, like finding the shortest path, i.e., the smallest number of edges that lead from one vertex to another, can become challenging on networks of this size. Even worse are harder problems, like finding the smallest set of vertices that cover all edges, for which no efficient, i.e., polynomial-time algorithm exists, unless verification in polynomial time is equivalent to solving in polynomial time (i.e., $P = NP$).

The efficiency of an algorithm is typically measured in terms of how long it takes to compute a solution, depending on the size of the input. Most commonly, this running time considers the worst case, yielding a reliable bound that holds for all inputs. For many interesting problems, including Karp's famous 21 NP-complete problems [Kar72], it is generally assumed that no algorithms exist that can solve them efficiently, i.e., in running time polynomial in the input size. Thus, computing a solution is practically intractable, in the worst case.

Therefore, it comes as a surprise that recent practical observations indicate that such problems can often be solved rather efficiently, yielding a huge gap

between empirical measurements and theoretical bounds. For one of the most fundamental NP-complete problems, *Boolean satisfiability*, solvers have proven to be remarkably efficient in practice [GV21]. Similar observations have been made in the context of complex networks, where NP-complete problems such as the classical *vertex cover* problem, can be solved in a matter of seconds, even if input and solution contain millions of vertices and edges [AI16]. Furthermore, this contrast has been repeatedly highlighted by the results of recent *PACE* challenges, which are set up to encourage the development of practical algorithms for notoriously hard problems [Del+18; DFH19; Kel+21].

A lot of research has focused on this discrepancy between theory and practice, as bridging this knowledge gap has several advantages. The first is educational. If we can prove that certain algorithms perform better than others in practice, people can make more proficient choices when using algorithms for practical purposes. Secondly, understanding the underlying cause for why algorithms perform well on certain networks may allow us to further exploit this behavior in order to develop even better algorithms.

Among various approaches to bridging the theory-practice gap the basic idea is typically to consider differences between the encountered instances. Theoretical bounds are often obtained by designing worst-case instances. However, real-world networks rarely resemble the worst case. In the context of *parameterized complexity*, this is acknowledged by finding parameters, i.e., quantifiable properties of input or solution, which yield better running times if they fulfill certain criteria, like being sufficiently small [Cyg+15]. Another approach is *smoothed analysis*, where worst-case inputs are considered after applying small random perturbations, motivated by the fact that real-world data is often noisy and thus unlikely to reflect the worst case [ST09]. A similar method is *average-case analysis*, which is the focus of this thesis. Instead of considering all possible inputs, including the worst case, the analysis is restricted to instances that are drawn from certain probability distributions. This allows us to focus on instances whose properties match what has been observed in practice.

Unfortunately, capturing real-world networks from a theoretical point of view is far from trivial. Typically, we can observe the relations between entities and reconstruct the network using these observations, but we are missing the knowledge of where the relations originate from. For example, we may not know why specific friendships in a social network have formed or why certain proteins interact. Consequently, we do not know a mathematical model describing how

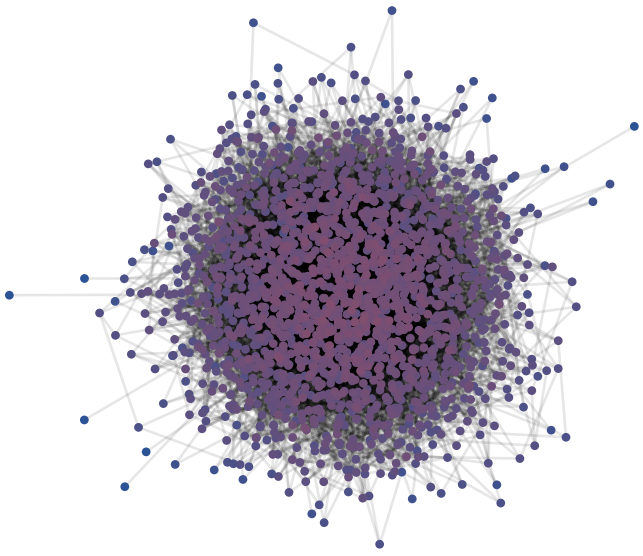


Figure 1.1: An Erdős-Rényi random graph with roughly 2000 vertices and an average degree of about 8. Colors represent the vertex degrees from low (blue) to high (red, not realized in this figure). They are consistent throughout all figures in this section.

edges in a real-world network form, which we could then analyze theoretically. In fact, such a model may not even exist. However, over time people have come up with their own models to represent real-world graphs as closely as possible.

Probably the most famous graph model are *Erdős-Rényi random graphs*, where, given a set of vertices, all vertex pairs are independently connected by an edge with the same probability [ER59; Gil59]. On the one hand, the model is mathematically very accessible, due to its simplicity. We refer the reader to the book by Bollobás for a comprehensive description of the model and its properties [Bol01]. On the other hand, the properties of the generated networks do not really match the above mentioned practical observations. In particular, since all edges exist independently with the same probability, all vertices have the same expected degree, see Figure 1.1. However, many real-world networks feature a *heterogeneous degree distribution*, where most vertices have similar, small degree but there are also a few vertices that have very high degree. In social networks, for example, most people have only few connections, while there are a couple of celebrities that are connected to a lot of people. Mathematically, such a degree

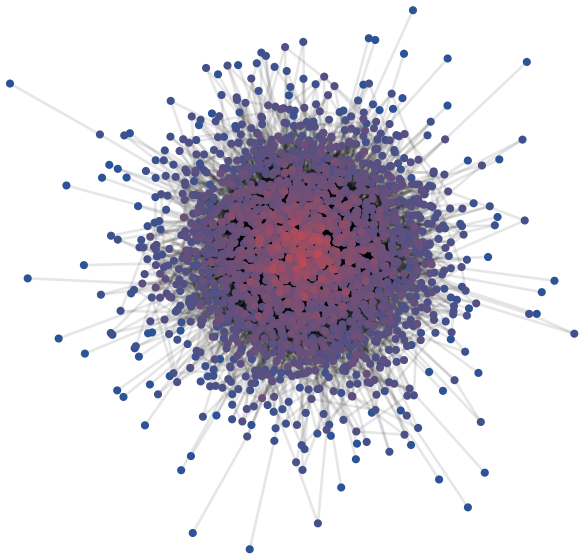


Figure 1.2: A Chung-Lu random graph with roughly 2000 vertices and an average degree of about 8. The power-law exponent of the degree distribution is approximately 2.5. Colors match the assignment in the previous figure. However, here they move further into the red, since there are a few vertices of higher degree.

distribution often resembles a *power law* [Art+20; Ser+21; Voi+19]. That is, the number of vertices in the network that have a degree of k is proportional to $k^{-\beta}$, where β is called the *power-law exponent*. In order to represent networks with heterogeneous degree distributions, other models have been proposed.

The *Chung-Lu model* allows for more diverse distributions, by assigning each vertex a *weight* and sampling an edge between two vertices with a probability that is proportional to the product of their weights [ACL01; CL02a; CL02b]. The expected degrees of the vertices then match the corresponding weights in the generated graph. Thus, if the weights follow a power-law distribution, so does the resulting degree sequence, in expectation; see Figure 1.2. Another model that allows for power-law degree distributions is the *Barabási-Albert model*. It follows a preferential attachment mechanism, where a graph is generated incrementally [BA99]. Starting with a simple graph, in each step a new vertex is added and connected to a fixed number of prior vertices, where the probability for an edge to form is proportional to their degrees, i.e., new vertices prefer

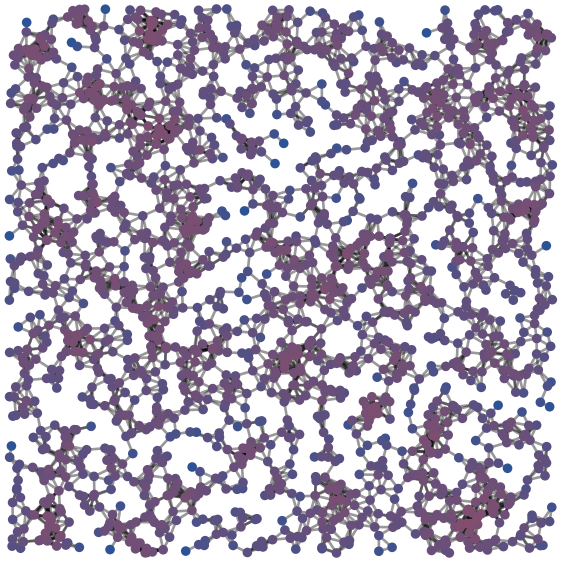


Figure 1.3: A Euclidean random graph with roughly 2000 vertices and an average degree of about 8. Colors match the assignment in the previous figures.

existing ones that have a high degree. For an overview of this and further such graph models, we refer to the book by Frieze and Karoński [FK15b] and the one by van der Hofstad [Hof16]. A drawback of these models is that edges form independently of each other, which leads to a clustering coefficient that vanishes with increasing graph size (see, e.g.,[Fou15; KE02]). However, in many real-world networks the clustering is rather high [New01; WS98]. In social networks, for example, two people are more likely to be friends, if they have a common friend. Thus, relations are not independent, which leads to clustering. One approach to obtain a network model with high clustering is to introduce geometry.

Euclidean random graphs (typically referred to as *random geometric graphs*) are obtained by distributing points uniformly at random within the unit hypercube and connecting any two of them, if and only if their Euclidean distance is below a certain threshold R. In the two-dimensional case, the points are distributed in the unit square $[0, 1]^2$, as illustrated in Figure 1.3, and one can imagine a disk of radius R around each point, which contains all other points that it is adjacent to. For a more detailed description of such graphs and their properties, we refer to the book by Penrose [Pen03]. Note that, given a set of points, we can set $R = 1$

and scale the ground space accordingly, while still obtaining the same graph. Consequently, such graphs can also be seen as randomly generated *unit disk graphs* (see, e.g., [Fis04]). In such networks the clustering coefficient does not vanish as the graph size increases. Instead, it is a constant that only depends on the dimension of the ground space [DC02]. To get an intuition for why this is the case, imagine two points that have a common neighbor. This means that they both lie within a disk of radius R centered at this neighbor. Consequently, the distance between them cannot be arbitrarily large (it is bounded by the diameter of the disk), making it more likely that they are adjacent themselves. Thus, the edges are not independent in this graph model and we obtain clustering. However, the geometry has another effect on the structure of the graph. Since the points are distributed uniformly at random, most points are expected to contain the same number of points in their neighborhood disks, leading to a rather homogeneous degree distribution.

To obtain networks that have clustering *and* a heterogeneous degree distribution, we can utilize the very same process but in the *hyperbolic geometry* instead of the Euclidean one. Research has shown that several real-world networks, like the internet, the network of international trade, and social networks, fit well into the *hyperbolic plane* (a two-dimensional surface on which space expands exponentially fast), which means that we can map the vertices in the graph to points in a hyperbolic disk, such that path lengths between vertices roughly match the hyperbolic distances between the corresponding points [BPK10; Gar+16; VS14].

Krioukov et al. formalized this relationship between complex networks and hyperbolic geometry with the introduction of the *hyperbolic random graph model* [Kri+10]. Instead of taking a graph and trying to map its vertices to points in the hyperbolic plane, points are first distributed at random in a disk of radius R in this plane, and a graph is generated by connecting any two points with an edge if their hyperbolic distance is below a threshold that matches the disk radius R. See Figure 1.4 for an exemplary illustration. The resulting graphs feature a power-law degree distribution [GPP12]. In fact, it was shown that hyperbolic random graphs are, in a sense, a variant of the above mentioned Chung-Lu model with the addition of dependencies between edges due to the underlying geometry [BFM15; Fou15]. Consequently, hyperbolic random graphs feature a non-vanishing clustering coefficient [GPP12]. In addition, they also have a small diameter [FK18; KM15; MS19], meaning they exhibit the *small-world phenomenon* describing that graph distances are typically short in practice [Bac+12; TM69;

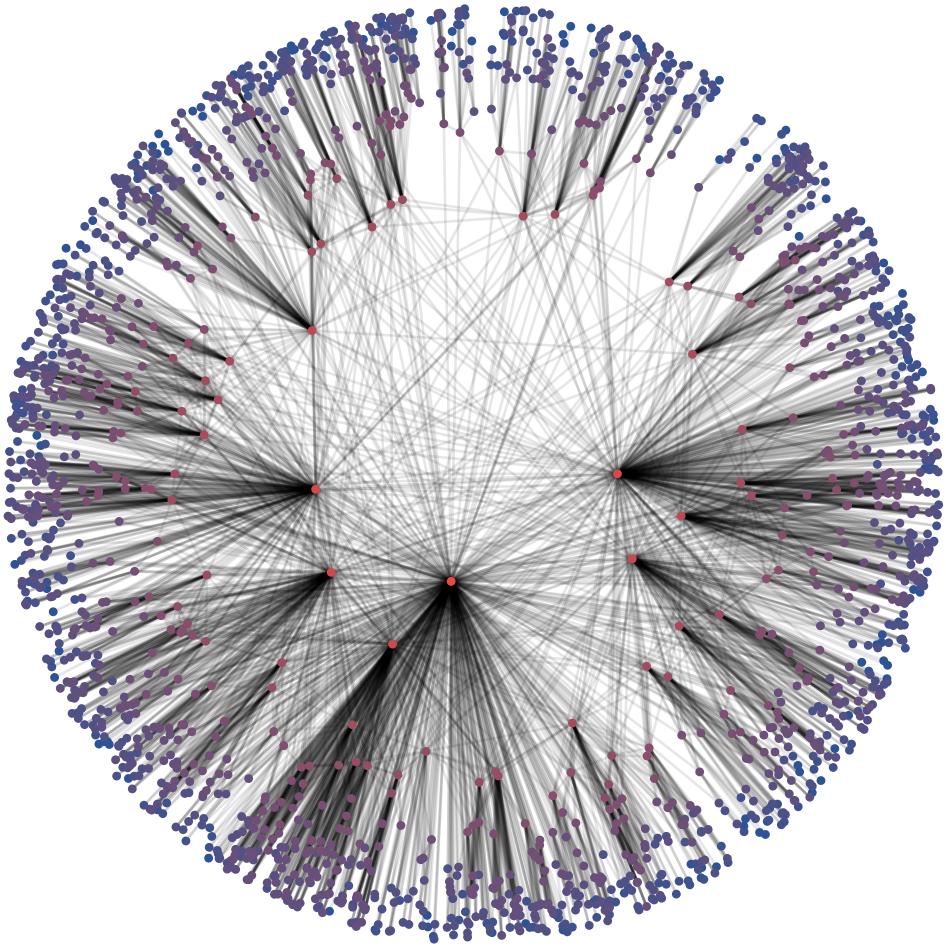


Figure 1.4: A hyperbolic random graph with roughly 2000 vertices and an average degree of about 8. The power-law exponent of the degree distribution is approximately 2.5. Colors match the assignment in the previous figures.

WS98]. Since hyperbolic random graphs feature all of these properties, which are often collected under the term *scale-freeness*, they are a good representation of many real-world networks where the same properties have been observed, including the internet, social networks, as well as protein-protein interaction networks [AB02; Alb05; Dor10].

As a consequence, hyperbolic random graphs are a promising model to be studied in the context of average-case complexity, in order to explain why algorithms tend to perform much better on real-world graphs than in the worst case. Not only do the generated graphs resemble practical instances well, the model is also conceptually simple enough such that it is accessible from a mathematical point of view. We note that, of course, the generative process of hyperbolic random graphs does not resemble *how* real-world networks form. People in a social network do not bond over being close in a hyperbolic disk. The model should rather be seen as a *tool* to facilitate the analysis of processes on real-world graphs.

Contribution and Outline

Throughout this thesis our goal is to support the validity of modelling networks using an underlying hyperbolic geometry for the purpose of average-case analysis. To this end, we consider several algorithmic problems that feature a gap between theoretical and practical understanding and analyze them using hyperbolic graph models. That is, each research effort is motivated by prior empirical observations, which we aim to explain theoretically. As a byproduct of our analyses we derive several predictions the model makes about network properties, which we empirically demonstrate to translate back to real-world networks. Eventually, we use these findings to further improve the performance of an existing algorithm on practical instances. The thesis is structured as follows.

In Chapter 2, we give a brief overview of the basics of graph theory and a more comprehensive introduction to probability theory, which is heavily used throughout the thesis due to the inherent randomness of the considered network model. In particular, we explain certain *concentration bounds* that are used to obtain meaningful statements. They include the *method of bounded differences*, as well as the *method of typically bounded differences* that, to the best of our knowledge, has not been applied in this context before.

In Chapter 3, we introduce networks with an underlying hyperbolic geometry

in greater detail. To this end, we start with an introduction to the hyperbolic plane and the *polar-coordinate model*, which we use to represent hyperbolic space. Additionally, we explain how the curvature (a parameterized property of the space) can be used to scale the plane. Afterwards, we introduce *hyperbolic unit disk graphs* as a graph class encompassing Euclidean unit disk graphs and *strongly hyperbolic unit disk graphs* on two ends of a spectrum that ranges from grid-like to hierarchical structures in sparse networks. We then continue with a detailed description of the hyperbolic random graph model, which we identify as a special case of hyperbolic unit disk graphs.

Chapter 4 provides the first insights into how hyperbolic unit disk graphs can be used to theoretically explain an empirically observed phenomenon. There, we study *routing*, i.e., the task of efficiently forwarding messages through a network, which is at the heart of communication networks like the internet. We develop and analyze a routing scheme that utilizes the hierarchies exhibited by strongly hyperbolic unit disk graphs and show that greedy routing can be implemented efficiently on such graphs. For the special case of hyperbolic random graphs, the scheme improves below existing performance lower bounds. There, the worst-case stretch, i.e., the deviation between routed paths and shortest paths in the graph, is at most 3 and at each vertex the scheme, asymptotically almost surely, stores $O(\log(n)^4)$ bits and makes routing decisions in $O(\log(n)^2)$ time. Further, our empirical analysis indicates that these results translate well to real-world networks.

In Chapter 5, we consider the breadth-first search (BFS), another fundamental path finding algorithm. A standard approach to accelerating this method is to consider a bidirectional version, which explores the graph from the start and the destination, simultaneously. In practice this strategy performs particularly well on scale-free real-world networks, as observed by Borassi and Natale [BN16]. To explain this behavior, we analyze the running time of the bidirectional BFS on hyperbolic random graphs and prove that it is $\tilde{O}\big(n^{2-1/\alpha} + n^{1/(2\alpha)} + \deg_{\max}(G)\big)$ with high probability, where $\alpha \in (1/2, 1)$ controls the power-law exponent of the degree distribution, and $\deg_{\max}(G)$ is the maximum degree of the graph. This bound is sublinear, improving the obvious worst-case linear bound. We note that, although our analysis depends on the underlying geometry, the algorithm itself is oblivious to it.

In Chapter 6, we turn our attention to the vertex cover problem, which is one of Karp's famous 21 NP-complete problems [Kar72]. Despite its computational

hardness, recent experiments by Akiba and Iwata [AI16] suggest that on many real-world networks the run time to find a solution is way smaller than even the best known approaches in the field of parameterized complexity can explain. We link these observations to the heterogeneity of the degree distribution and high clustering that are often observed in real-world networks, by analyzing how a branch-and-reduce algorithm performs on hyperbolic random graphs. In fact, we are able to show that the vertex cover problem on hyperbolic random graphs can be solved in polynomial time, with high probability. The proof relies on interesting structural properties of hyperbolic random graphs, which we empirically find to be present in real-world networks as well.

In Chapter 7, we further investigate the vertex cover problem with a focus on its approximability. While the above mentioned results show that an optimal vertex cover can be computed in polynomial time, the degree of the polynomial is unknown and for large networks, even a quadratic running time may be too much to be practically tractable. One approach to dealing with the computational hardness of such a problem is to trade the qualitative performance of an algorithm (allowing non-optimal outputs) for an improved running time. When it comes to understanding this trade-off for the vertex cover problem, there is, again, a gap between theory and practice. On the one hand, it is known that it is NP-hard to approximate a minimum vertex cover within a factor of $\sqrt{2}$. On the other hand, a simple greedy algorithm yields close to optimal approximations in practice. We close the theory-practice gap by providing an algorithm that efficiently computes nearly optimal vertex cover approximations on hyperbolic random graphs. More precisely, our algorithm computes a $(1 + o(1))$-approximation, asymptotically almost surely, and has a running time of $O\big(m \log(n)\big)$. The proposed algorithm is an adaptation of the successful greedy approach, enhanced with a procedure that improves on parts of the graph where greedy is not optimal. This makes it possible to introduce a parameter that can be used to tune the trade-off between approximation performance and running time. Our empirical evaluation on real-world networks shows that this allows for improving over the near-optimal results of the greedy approach.

We conclude our findings in Chapter 8 and present several directions in which our research can be extended, including extensions to *noisy settings* that allow for slack in the otherwise strict generative rules of hyperbolic unit disk graphs, on to extensions of the model to other graph types such as trees, and further generalizations regarding the dimensionality of the considered space.

2 Preliminaries

2.1 Graph Theory

In this thesis, we study how algorithms behave on certain *graphs* or *networks*. They provide a mathematical model to study relations between entities in the real world, like friendships among people in a social networks or links between routers in the internet. In the following, we give a brief overview of the basic graph theoretic concepts and the notation used in the thesis. For a more detailed introduction to graph theory, we refer the reader to the book by Diestel [Die17].

Formally, a graph $G = (V, E)$ models the entities as a finite set of *vertices* V and the relations as a set of *edges* $E \subseteq V \times V$. We use $|\cdot|$ to denote cardinalities. In particular, the number of vertices and edges in a graph are denoted by $n = |V|$ and $m = |E|$, respectively. When using big-O notation, the asymptotics are assumed to be in n and we use $\tilde{O}(\cdot)$ to suppress polylogarithmic factors in n.

Two vertices $u, v \in V$ are *adjacent*, if they are connected by an edge $e = \{u, v\} \in E$. We say that the edge e is *incident* to u and v. Note that we consider *undirected* graphs, as $\{u, v\} = \{v, u\}$. The *neighborhood* $N(v)$ of a vertex v is the set of vertices adjacent to v. The size of the neighborhood is called the *degree* of v and is denoted by $\deg(v) = |N(v)|$. Moreover, the *clustering coefficient of a vertex* v denotes the probability for two randomly chosen neighbors of v to be adjacent and the *clustering coefficient of a graph* denotes the average of the coefficients of all vertices.

A vertex u is indirectly connected to a vertex v outside of its neighborhood if there exists a *path* from u to v, which is a sequence $(u, \ldots, v)$ of vertices where every vertex is adjacent to its successor. The *length* of the path is one less than the length of the sequence. The *distance* between u and v is the length of a shortest path between them and the *diameter* of G, written as $\operatorname{diam}(G)$, denotes the maximum distance among vertices in G. Unless stated otherwise, we consider graphs that are *connected*, meaning there exists a path between all vertex pairs.

For a subset $S \subseteq V$, we use $G[S]$ to denote the *induced subgraph* of G obtained by removing all vertices in $V \setminus S$ together with their incident edges.

2.2 Probability Theory

Throughout the thesis, we analyze graphs that are generated using a mechanism involving randomness. Intuitively, we consider all possible graphs that may be generated by the mechanism and observe their properties or how an algorithm performs on them. However, since the number of graphs the mechanism may produce is infinite, we instead make probabilistic predictions about them, which are typically of the form: "When a graph is generated using the mechanism, it is very likely that it has a certain property." In the following, we give an overview of how this can be formalized. For a more comprehensive introduction to probability theory, we refer the reader to book by Ross [Ros19].

A random experiment can be described by a *probability space*, which is a tuple (Ω, Σ, P) that consists of a *sample space* Ω, containing all possible outcomes of the experiment, a σ-algebra Σ containing sets of outcomes in the sample space, called *events*, and a *probability function* P assigning each event a probability, i.e., a number in $[0, 1]$. For example, Ω could contain all graphs on n vertices and the event "the graph is connected" would be represented by a set in Σ that contains all n-vertex graphs where each vertex pair is connected by a path. We denote the probability for an event $A \in \Sigma$ to occur by $\Pr[A]$. The complementary event of A, denoted by $\neg A$, represents the case where A does not occur. Its probability is given by $\Pr[\neg A] = 1 - \Pr[A]$. Given another event $B \in \Sigma$, the probability that A *or* B occurs is given by

$$\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B].$$

Since $\Pr[A \cap B]$ is non-negative, we have $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$. In general, for a subset $\Sigma' \subseteq \Sigma$ the *union bound* states that $\Pr\left[\bigcup_{A \in \Sigma'} A\right] \leq \sum_{A \in \Sigma'} \Pr[A]$.

The probability that the event A occurs, under the assumption that an event B has already occurred is called the *conditional probability of A given B* and is denoted by $\Pr\left[A \mid B\right] = \Pr[A \cap B]/\Pr[B]$. The events A and B are said to be *independent* if

$$\Pr[A \cap B] = \Pr[A] \cdot \Pr[B], \text{ or equivalently, if } \Pr\left[A \mid B\right] = \Pr[A].$$

Intuitively, this means that whether B occurs has no impact on whether A does.

A *random variable* describes a value that depends on the outcome of a random experiment. We could define a random variable whose value is 1 if the generated

graph is connected and 0 if it is not. Another random variable is the number of vertices in the graph that have a degree of 1. Formally, we consider a random variable X as a function that assigns each possible outcome a value in a measurable set S, i.e., $X \colon \Omega \to S$. The probability that X takes on a certain value $x \in S$, is described using a *cumulative distribution function* $F_X \colon S \to [0, 1]$, which is given by

$$F_X(x) = \Pr[X \leq x].$$

Given two random variables X and Y, their *joint distribution function* is given by

$$F_{X,Y}(x, y) = \Pr\big[X \leq x \wedge Y \leq y\big].$$

We say that X and Y are *independent* if $F_{X,Y}(x, y) = F_X(x) \cdot F_Y(y)$.

If X maps to a countable set S, we call it a *discrete random variable*. Its distribution is then often described using a *probability mass function* f_X, which is defined as

$$f_X(x) = \Pr[X = x] = \Pr\big[\{\omega \in \Omega \mid X(\omega) = x\}\big].$$

This function relates to the cumulative distribution function as

$$F_X(x) = \sum_{\substack{x' \in S, \\ x' \leq x}} f_X(x').$$

The *expected value* or *expectation* $\mathbb{E}[X]$ intuitively describes the average outcome for an infinite amount of trials. It is defined as

$$\begin{aligned} \mathbb{E}[X] &= \sum_{x \in S} x \cdot f_X(x) \\ &= \sum_{x \in S} x \cdot \Pr[X = x]. \end{aligned}$$

Moreover, the expected value of X *conditioned on an event* $A \in \Sigma$ with non-zero probability (i.e., $\Pr[A] > 0$) is

$$\mathbb{E}\big[X \mid A\big] = \sum_{x \in S} x \cdot \Pr\big[X = x \mid A\big].$$

This allows us to express the expected value of a random variable X as a sum over a partition Q of the sample space Ω, via the *law of total expectation*, which states that

$$\mathbb{E}[X] = \sum_{A \in Q} \mathbb{E}\big[X \mid A\big] \cdot \Pr[A]. \tag{2.1}$$

On the other hand, we consider *continuous random variables* $X \colon \Omega \to S$, where $S \subseteq \mathbb{R}$. Then, f_X is typically called the *probability density function*. In this case, the conditional distribution of X given an event $A \in \Sigma$ (with $\Pr[A] > 0$) is defined as

$$f_{X|A}(x) = \begin{cases} \frac{f_X(x)}{\Pr[A]}, & x \in A, \\ 0, & x \notin A. \end{cases}$$

Moreover, the *joint probability density function* of two continuous random variables X and Y is given by

$$\begin{aligned} f_{X,Y}(x,y) &= \frac{F_{X,Y}(x,y)\,\mathrm{d}^2}{\mathrm{d}x\,\mathrm{d}y} \\ &= f_{X|Y=y}(x) \cdot f_Y(y). \end{aligned}$$

Analogous to the definition for discrete random variables, the expected value of a continuous random variable X is

$$\mathbb{E}[X] = \int_{x \in S} x \cdot f_X(x)\,\mathrm{d}x,$$

and the conditional expectation given an event $A \in \Sigma$ with non-zero probability is defined as

$$\mathbb{E}\big[X \mid A\big] = \int_{x \in S} x \cdot f_{X|A}(x)\,\mathrm{d}x.$$

Often, determining the expected value of a random variable is not sufficient to obtain meaningful statements. When considering random experiments related to graphs, we therefore classify events depending on how likely they are to occur. We say that an event happens *with high probability* and *asymptotically almost surely*, if it occurs with probability $1 - \mathrm{O}\big(1/n\big)$ and $1 - \mathrm{o}(1)$, respectively.

2.2.1 Chernoff Bounds

To show that certain random variables are concentrated around their expectation, i.e., with high probability the outcome does not deviate much from the expected value, we regularly use the following Chernoff bounds.

Theorem 2.1 (Chernoff Bounds [MU05, Theorems 4.4 and 4.5])**.** *Let $X_1, \dots, X_n$ be independent random variables taking values in the set $\{0, 1\}$ and let X be their sum. Then,*

$$\begin{aligned} \Pr[X \geq t] &\leq 2^{-t} && \text{for } t \geq 6\mathbb{E}[X], \\ \Pr\big[X \geq (1+\varepsilon)\mathbb{E}[X]\big] &\leq e^{-\varepsilon^2/3 \cdot \mathbb{E}[X]} && \text{for } \varepsilon \in (0, 1], \text{ and} \\ \Pr\big[X \leq (1-\varepsilon)\mathbb{E}[X]\big] &\leq e^{-\varepsilon^2/2 \cdot \mathbb{E}[X]} && \text{for } \varepsilon \in (0, 1). \end{aligned}$$

Usually, it suffices to show that a random variable does not exceed a certain upper bound or drop below a lower bound with high probability. The following corollaries show that (sufficiently large) upper and lower bounds on the expected value suffice to obtain concentration.

Corollary 2.2. *Let $X_1, \dots, X_n$ be independent random variables taking values in the set $\{0, 1\}$, let X be their sum, and let $f(n)$ be an upper bound on $\mathbb{E}[X]$. Then, for all $\varepsilon \in (0, 1)$ it holds that*

$$\Pr\big[X \geq (1+\varepsilon)f(n)\big] \leq e^{-\varepsilon^2/3 \cdot f(n)}.$$

Proof. Consider a random variable X' with $\mathbb{E}[X'] = f(n)$ such that $X \leq X'$ for every outcome. Note that X' exists as $f(n) \geq \mathbb{E}[X]$. Since $X \leq X'$, it holds that

$$\Pr\big[X \geq (1+\varepsilon)f(n)\big] \leq \Pr\big[X' \geq (1+\varepsilon)f(n)\big] = \Pr\Big[X' \geq (1+\varepsilon)\mathbb{E}\big[X'\big]\Big].$$

Using Theorem 2.1 we can derive that

$$\begin{aligned} \Pr\Big[X' \geq (1+\varepsilon)\mathbb{E}\big[X'\big]\Big] &\leq e^{-\varepsilon^2/3 \cdot \mathbb{E}[X']} \\ &= e^{-\varepsilon^2/3 \cdot f(n)}. \end{aligned}$$

□

In particular, the above corollary implies that X does not exceed an upper bound on the expected value $\mathbb{E}[X]$ with high probability, if this bound is sufficiently large.

Corollary 2.3. *Let $X_1, \dots, X_n$ be independent random variables taking values in the set $\{0, 1\}$ and let X be their sum. Further, let $f(n) = \Omega\big(\log(n)\big)$ be such that $\mathbb{E}[X] \leq f(n)$ and let $c \geq 0$ be a constant. Then, with probability $1 - \mathrm{O}(n^{-c})$ it holds that $X = \mathrm{O}\big(f(n)\big)$.*

Proof. We prove the statement by showing that the probability for the complementary event, i.e., X is more than a constant factor larger than $f(n)$, is $\mathrm{O}(n^{-c})$ for any $c \geq 0$. Since $\mathbb{E}[X] \leq f(n)$, we can choose a constant c_1 sufficiently large such that $c_1 f(n) \geq 6\mathbb{E}[X]$. Thus, by Theorem 2.1 it holds that

$$\Pr\big[X \geq c_1 f(n)\big] \leq 2^{-c_1 f(n)}.$$

Moreover, we have $f(n) = \Omega\big(\log n\big)$. Consequently, there exists another constant c_2 such that $f(n) \geq c_2 \log(n)$ for sufficiently large n. We obtain

$$\Pr\big[X \geq c_1 f(n)\big] \leq 2^{-c_1 c_2 \log(n)} \leq n^{-c_1 c_2}$$

for n sufficiently large. Then, choosing c_1 such that $c_1 > c/c_2$ yields the claim. □

Similarly, a super-logarithmic lower bound on $\mathbb{E}[X]$ implies that X does not fall below the bound with high probability.

Corollary 2.4. *Let $X_1, \dots, X_n$ be independent random variables taking values in the set $\{0, 1\}$ and let X be their sum. Further, let $f(n) = \omega\big(\log(n)\big)$ be such that $f(n) \leq \mathbb{E}[X]$ and let $c \geq 0$ be a constant. Then, with probability $1 - \mathrm{O}(n^{-c})$ it holds that $X = \Omega\big(f(n)\big)$.*

Proof. Analogous to the proof of Corollary 2.3 we prove the statement by showing that the probability for the complementary event, i.e., X is more than a constant factor smaller than $f(n)$, is $\mathrm{O}(n^{-c})$ for any $c \geq 0$. Let $\varepsilon \in (0, 1)$ be a constant. The following inequalities are obtained by first using the fact that $f(n) \leq \mathbb{E}[X]$, applying the third statement of Theorem 2.1, again applying $f(n) \leq \mathbb{E}[X]$, and finally using $f(n) = \omega\big(\log(n)\big)$:

$$\begin{aligned}
\Pr\big[X \leq (1-\varepsilon) f(n)\big] &\leq \Pr\big[X \leq (1-\varepsilon)\mathbb{E}[X]\big] \\
&\leq e^{-\varepsilon^2/2 \cdot \mathbb{E}[X]} \\
&\leq e^{-\varepsilon^2/2 \cdot f(n)} \\
&= e^{-\varepsilon^2/2 \cdot \omega\left(\log(n)\right)} = n^{-\omega(1)}.
\end{aligned}$$

□

2.2.2 Bounded Differences

While the Chernoff bound considers the sum of indicator random variables, we also have to deal with different functions of random variables. In this case tight bounds on the probability that the function deviates a lot from its expected value can be obtained using *the method of bounded differences*.

Let $X_1, \dots, X_n$ be independent random variables taking values in a set S. We say that a function $f\colon S^n \to \mathbb{R}$ satisfies the *bounded differences condition* if for all $i \in [n]$ there exists a $\Delta_i \geq 0$ such that

$$|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leq \Delta_i, \tag{2.2}$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in S^n$ that differ only in the i-th component.

Theorem 2.5 (Method of Bounded Differences [DP12, Corollary 5.2])**.** *Let $X_1, \dots, X_n$ be independent random variables taking values in a set S and let $f\colon S^n \to \mathbb{R}$ be a function that satisfies the bounded differences condition with parameters $\Delta_i \geq 0$ for $i \in [n]$. Then for $\Delta = \sum_{i\in[n]} \Delta_i^2$ it holds that*

$$\Pr\Big[f > \mathbb{E}\big[f\big] + t\Big] \leq e^{-2t^2/\Delta}.$$

As before, we are usually interested in showing that a random variable does not exceed a certain upper bound with high probability. Analogously to the Chernoff bound in Corollary 2.2, one can show that, again, an upper bound on the expected value suffices to show concentration.

Corollary 2.6. *Let $X_1, \dots, X_n$ be independent random variables taking values in a set S and let $f\colon S^n \to \mathbb{R}$ be a function that satisfies the bounded differences condition with parameters $\Delta_i \geq 0$ for $i \in [n]$. If $g(n)$ is an upper bound on $\mathbb{E}\big[f\big]$ then for $\Delta = \sum_{i\in[n]} \Delta_i^2$ and $c \geq 1$ it holds that*

$$\Pr\big[f > cg(n)\big] \leq e^{-2((c-1)g(n))^2/\Delta}.$$

Proof. Let $h(n) \geq 0$ be a function with $\hat{f} = f + h(n)$ such that $\mathbb{E}\big[\hat{f}\big] = g(n)$. Note that $h(n)$ exists since $g(n) \geq \mathbb{E}\big[f\big]$. As a consequence, we have $f \leq \hat{f}$ for all outcomes of $X_1, \dots, X_n$ and it holds that

$$\big|\hat{f}(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}')\big| = \big|f(\boldsymbol{x}) + h(n) - f(\boldsymbol{x}') - h(n)\big| = \big|f(\boldsymbol{x}) - f(\boldsymbol{x}')\big|,$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in S^n$. Consequently, $\hat{f}$ satisfies the bounded differences condition with the same parameters Δ_i as f. Since $f \leq \hat{f}$ it holds that

$$\Pr\left[f > cg(n)\right] \leq \Pr\left[\hat{f} > cg(n)\right] = \Pr\left[\hat{f} > c\mathbb{E}\left[\hat{f}\right]\right].$$

Choosing $t = (c-1)\mathbb{E}\left[\hat{f}\right]$ allows us to apply Theorem 2.5 to conclude that

$$\Pr\left[\hat{f} > c\mathbb{E}\left[\hat{f}\right]\right] = \Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t\right] \leq e^{-2((c-1)\mathbb{E}[\hat{f}])^2/\Delta} = e^{-2((c-1)g(n))^2/\Delta}. \square$$

2.2.3 Typical Bounded Differences

A disadvantage of the method of bounded differences is that one has to consider the worst possible change in f when changing one variable and the resulting bound becomes worse the larger this change. A way to overcome this issue is to consider the method of *typical* bounded differences instead. Intuitively, it allows us to milden the effect of the change in the worst case, if it is sufficiently unlikely, and to focus on the typical cases where the change should be small, instead. Formally, we say that a function $f\colon S^n \to \mathbb{R}$ satisfies the *typical bounded differences condition* with respect to an event $A \subseteq S^n$ if for all $i \in [n]$ there exists a $\Delta_i^A \leq \Delta_i$ such that

$$|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leq \begin{cases} \Delta_i^A, & \text{if } \boldsymbol{x} \in A, \\ \Delta_i, & \text{otherwise,} \end{cases} \tag{2.3}$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in S^n$ that differ only in the i-th component.

Theorem 2.7 (Method of Typical Bounded Differences, [War16, Theorem 2][1])**.** *Let $X_1, \dots, X_n$ be independent random variables taking values in a set S and let $A \subseteq S^n$ be an event. Furthermore, let $f\colon S^n \to \mathbb{R}$ be a function that satisfies the typical bounded differences condition with respect to A and with parameters $\Delta_i^A \leq \Delta_i$ for $i \in [n]$. Then for all $\varepsilon_1, \dots, \varepsilon_n \in (0, 1]$ there exists an event B satisfying $\neg B \subseteq A$ and $\Pr[B] \leq \Pr[\neg A] \cdot \sum_{i\in[n]} 1/\varepsilon_i$, such that for $\Delta = \sum_{i\in[n]} (\Delta_i^A + \varepsilon_i(\Delta_i - \Delta_i^A))^2$ and $t \geq 0$ it holds that*

$$\Pr\left[f > \mathbb{E}\left[f\right] + t \wedge \neg B\right] \leq e^{-t^2/(2\Delta)}.$$

1 We state a slightly simplified version in order to facilitate understandability. The original theorem allows for the random variables $X_1, \dots, X_n$ to take values in different sets.

Intuitively, the choice of the values for ε_i has two effects. On the one hand, choosing ε_i small allows us to compensate for a potentially large worst-case change Δ_i. On the other hand, this also increases the bound on the probability of the event B that represents the atypical case. However, in that case one can still obtain meaningful bounds if the typical event A occurs with high enough probability. Again, it is usually sufficient to show that the function f does not exceed an upper bound on its expected value with high probability. The proof of the following corollary is analogous to the one of Corollary 2.6.

Corollary 2.8. *Let $X_1, \dots, X_n$ be independent random variables taking values in a set S and let $A \subseteq S^n$ be an event. Furthermore, let $f\colon S^n \to \mathbb{R}$ be a function that satisfies the typical bounded differences condition with respect to A and with parameters $\Delta_i^A \leq \Delta_i$ for $i \in [n]$, and let $g(n)$ be an upper bound on $\mathbb{E}[f]$. Then for all $\varepsilon_1, \dots, \varepsilon_n \in (0,1]$, $\Delta = \sum_{i\in[n]}(\Delta_i^A + \varepsilon_i(\Delta_i - \Delta_i^A))^2$, and $c \geq 1$ it holds that*

$$\Pr[f > cg(n)] \leq e^{-((c-1)g(n))^2/(2\Delta)} + \Pr[\neg A] \cdot \sum_{i\in[n]} 1/\varepsilon_i.$$

Proof. Let $h(n) \geq 0$ be a function with $\hat{f} = f + h(n)$ such that $\mathbb{E}[\hat{f}] = g(n)$. Note that $h(n)$ exists since $g(n) \geq \mathbb{E}[f]$. As a consequence, we have $f \leq \hat{f}$ for all outcomes of $X_1, \dots, X_n$ and it holds that

$$\begin{aligned} |\hat{f}(\boldsymbol{x}) - \hat{f}(\boldsymbol{x}')| &= |f(\boldsymbol{x}) + h(n) - f(\boldsymbol{x}') - h(n)| \\ &= |f(\boldsymbol{x}) - f(\boldsymbol{x}')|, \end{aligned}$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in S^n$. Consequently, $\hat{f}$ satisfies the typical bounded differences condition with respect to A with the same parameters $\Delta_i^A \leq \Delta_i$ as f. Since $f \leq \hat{f}$, it holds that

$$\begin{aligned} \Pr[f > cg(n)] &\leq \Pr\left[\hat{f} > cg(n)\right] \\ &= \Pr\left[\hat{f} > c\mathbb{E}[\hat{f}]\right]. \end{aligned}$$

By choosing $t = (c-1)\mathbb{E}[\hat{f}]$ this can be written as

$$\Pr\left[\hat{f} > c\mathbb{E}[\hat{f}]\right] = \Pr\left[\hat{f} > \mathbb{E}[\hat{f}] + t\right].$$

Theorem 2.7 now guarantees the existence of an event B with $\neg B \subseteq A$ and

$$\Pr[B] \leq \Pr[\neg A] \cdot \sum_{i \in [n]} 1/\varepsilon_i,$$

such that

$$\Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t \wedge \neg B\right] \leq e^{-t^2/(2\Delta)}.$$

To bound $\Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t\right]$ we apply the law of total probability and consider the events B and $\neg B$ separately

$$\begin{aligned}\Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t\right] = \Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t \;\middle|\; \neg B\right] \cdot \Pr[\neg B] \\ + \Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t \;\middle|\; B\right] \cdot \Pr[B].\end{aligned}$$

The first part of the sum can be simplified using the definition of conditional probabilities. Moreover, it holds that $\Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t \;\middle|\; B\right] \leq 1$. Thus, we can bound the above term by

$$\Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t\right] \leq \Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t \wedge \neg B\right] + \Pr[B].$$

Both remaining summands can now be bounded using the upper bounds that we previously obtained by applying Theorem 2.7, i.e.,

$$\Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t \wedge \neg B\right] \leq e^{-t^2/(2\Delta)} \quad \text{and} \quad \Pr[B] \leq \Pr[\neg A] \cdot \sum_{i \in [n]} 1/\varepsilon_i.$$

Then, it follows that

$$\Pr\left[\hat{f} > \mathbb{E}\left[\hat{f}\right] + t\right] \leq e^{-t^2/(2\Delta)} + \Pr[\neg A] \cdot \sum_{i \in [n]} 1/\varepsilon_i.$$

Finally, since t was chosen as $t = (c-1)\mathbb{E}\left[\hat{f}\right]$ and since $\mathbb{E}\left[\hat{f}\right] = g(n)$, we obtain the claimed bound. □

2.3 Useful Inequalities

Computations can often be simplified by making use of the fact that $1 \pm x$ can be closely approximated by $e^{\pm x}$ for small x. More precisely, we use the following lemmas, which have been derived previously using the Taylor approximation [Kro16].

Lemma 2.9 ([Kro16, Lemma 2.1])**.** *Let $x \in \mathbb{R}$. Then, $1 + x \leq e^x$.*

Lemma 2.10 ([Kro16, Corollary of Lemma 2.2])**.** *Let $x > 0$ with $x = \mathrm{o}(1)$. Then,*

$$1 - x \geq e^{-(1+\mathrm{o}(1))x}.$$

Proof. Consider $\varepsilon = \log(1/(1-x)) = \mathrm{o}(1)$ and note that $e^{-\varepsilon} = 1 - x$. To prove the claim, it suffices to show that $e^{-\varepsilon} \geq e^{-(1+\varepsilon)x}$. It is easy to see that

$$e^{-\varepsilon} = e^{-\left(1+\varepsilon-\frac{1+\varepsilon}{1+\varepsilon}\right)} = e^{-(1+\varepsilon)\left(1-\frac{1}{1+\varepsilon}\right)}.$$

By Lemma 2.9 we have $1 + \varepsilon \leq e^{\varepsilon}$ and, thus, $1/(1+\varepsilon) \geq e^{-\varepsilon}$. Since ε is chosen such that $e^{-\varepsilon} = 1 - x$, we can bound $1/(1+\varepsilon) \geq 1 - x$ in above equation, which yields

$$e^{-\varepsilon} \geq e^{-(1+\varepsilon)(1-(1-x))} = e^{-(1+\varepsilon)x}.$$ □

Moreover, the following lemma can often be used to simplify error terms.

Lemma 2.11 ([Kro16, Lemma 2.3])**.** *Let $x \in \mathbb{R}$ with $x = \pm\mathrm{o}(1)$ be given. Then,*

$$\frac{1}{1+x} = 1 - \Theta(x).$$

3 Networks with Underlying Hyperbolic Geometry

Hyperbolic and Euclidean geometry differ in several fundamental ways. Commonly, hyperbolic geometry is introduced as a variation of Euclidean geometry, where the parallel axiom is replaced by a different one that allows for a given line to have more than one parallel that goes through a given point. Another distinguishing property is that, in contrast to flat Euclidean space, hyperbolic space is negatively curved. For us, the most important difference is the rate at which space expands. In the Euclidean plane the area of a disk grows with its radius R as πR^2, i.e., polynomially in R. In contrast, this growth is described by $2\pi(\cosh(R) - 1)$, with $\cosh(R) = (e^R + e^{-R})/2$, in the hyperbolic plane. That is, space expands exponentially fast there.

To capture these properties and facilitate the study of hyperbolic space, various *models of hyperbolic geometry* have been developed over time. While most of them are used to represent hyperbolic space of arbitrary dimensionality, we focus on the two-dimensional hyperbolic plane in this thesis. Formally, this plane is described using an abstract two-dimensional surface that is equipped with the axioms of hyperbolic geometry. We refer the reader to the book by Ramsay and Richtmyer [RR95] for an in-depth introduction. Intuitively, the different models of hyperbolic geometry present different ways of addressing points in the abstract surface. Arguably the most famous model is the *Poincaré disk model*, where the infinite hyperbolic plane is mapped into the Euclidean unit disk. Other commonly used representations include the *upper-half-plane model* and the *hyperboloid model*. We refer to [RR95, Chapter 7] for an overview of several models.

3.1 The Polar-Coordinate Model

In this thesis, we work with the *polar-coordinate model*, which is also called the *native representation* (see, e.g.,[Kri+10]) or *Gans model* after David Gans, who (to the best of our knowledge) proposed it first [Gan66].

In the following, we introduce the polar-coordinate model and present certain

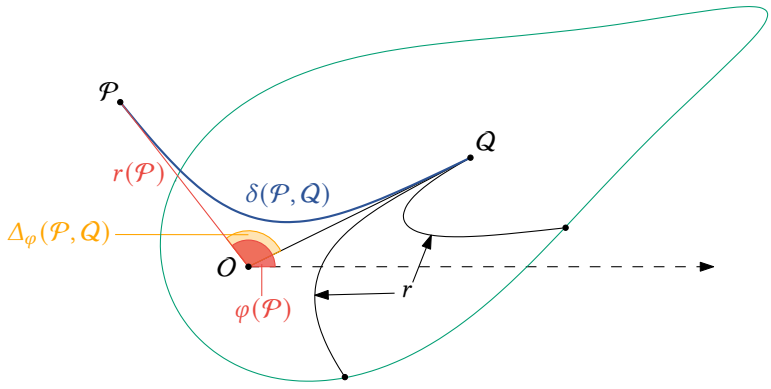


Figure 3.1: Visualization of the polar-coordinate model of the hyperbolic plane with curvature -1. Point $\mathcal{P}$ is identified using the radius $r(\mathcal{P})$ and angle $\varphi(\mathcal{P})$ (both red). The angular distance between $\mathcal{P}$ and $\mathcal{Q}$ is shown in orange. The line segment representing the hyperbolic distance between them is shown in blue. A circle (green) of radius r is centered at $\mathcal{Q}$.

properties of the hyperbolic plane that are relevant for our analyses. Throughout the thesis, we use calligraphic symbols to denote geometric objects. Consider the two-dimensional hyperbolic plane $\mathbb{H}^2_\zeta$ of negative curvature $K = -\zeta^2$ for a constant $\zeta > 0$. To address points in the plane, we use *polar coordinates*. After defining a designated *pole* or *origin* $\mathcal{O} \in \mathbb{H}^2_\zeta$ together with a *polar axis*, i.e., a reference ray starting at $\mathcal{O}$, a point $\mathcal{P} \in \mathbb{H}^2_\zeta$ is uniquely determined by a tuple $(r(\mathcal{P}), \varphi(\mathcal{P}))$, where $r(\mathcal{P})$ denotes the *radius* of $\mathcal{P}$, which is the hyperbolic distance to $\mathcal{O}$, and $\varphi(\mathcal{P})$ denotes *angle* (or *angular coordinate*) of $\mathcal{P}$, which is the angle between the reference ray and the ray from $\mathcal{O}$ through $\mathcal{P}$, measured in counterclockwise direction around $\mathcal{O}$. For visualizations, these coordinates are then interpreted as polar coordinates in the Euclidean plane; see Figure 3.1. Since hyperbolic space expands faster than Euclidean space, this projection leads to a distortion. Consequently, line segments bend towards the origin $\mathcal{O}$ and circles appear tear-drop shaped.

Intuitively, the curvature can be used to "scale" the space, i.e., it affects the hyperbolic distance between points, which can be determined using the *hyperbolic law of cosines*. For a triangle, it allows us to compute the length c of one side, given the lengths a, b of the other sides and the angle φ opposite of c. In

particular, it states that

$$\cosh(\zeta c) = \cosh(\zeta a)\cosh(\zeta b) - \sinh(\zeta a)\sinh(\zeta b)\cos(\varphi),$$

where $\sinh(x) = (e^x - e^{-x})/2$. To compute the *hyperbolic distance* between two points $\mathcal{P}, \mathcal{Q} \in \mathbb{H}^2_\zeta$, we consider the triangle defined by $\mathcal{P}, \mathcal{Q}$, and the origin $\mathcal{O}$. Two side lengths are given by the radii $r(\mathcal{P})$ and $r(\mathcal{Q})$, and the angle opposite of the third side, i.e., the side representing the hyperbolic distance between $\mathcal{P}$ and $\mathcal{Q}$ (blue in Figure 3.1), is the *angular distance* between the two points, i.e., the angle between the rays from the origin through $\mathcal{P}$ and $\mathcal{Q}$ (orange in Figure 3.1). Formally, the angular distance is defined as

$$\delta_\varphi(\mathcal{P}, \mathcal{Q}) = \pi - |\pi - |\varphi(\mathcal{P}) - \varphi(\mathcal{Q})||.$$

Then, the hyperbolic distance between $\mathcal{P}$ and $\mathcal{Q}$ is given by

$$\begin{aligned}\delta_{\mathbb{H}^2_\zeta}(\mathcal{P}, \mathcal{Q}) = \frac{1}{\zeta}\operatorname{acosh}\big(&\cosh(\zeta r(\mathcal{P}))\cosh(\zeta r(\mathcal{Q})) \\ &- \sinh(\zeta r(\mathcal{P}))\sinh(\zeta r(\mathcal{Q}))\cos(\delta_\varphi(\mathcal{P}, \mathcal{Q}))\big),\end{aligned}$$

where $\operatorname{acosh}(x) = \log(x + \sqrt{x^2 - 1})$ is the inverse of the hyperbolic cosine. Note how the angular coordinates make simple definitions cumbersome as angles are considered modulo 2π, leading to a case distinction depending on where the reference ray lies. Whenever possible, we implicitly assume that the reference ray is chosen such that we do not have to compute modulo 2π. Moreover, we can utilize two properties of the cosine function to simplify the above distance function. First, we can use the identity $\cos(x-y) = \cos(x)\cos(y) + \sin(x)\sin(y)$, from which we can derive

$$\cos(\pi - x) = \underbrace{\cos(\pi)}_{-1}\cos(x) + \underbrace{\sin(\pi)}_{0}\sin(x) = -\cos(x). \tag{3.1}$$

Second, we can make use of the fact that the cosine function is symmetric about the y-axis, which means $\cos(-x) = \cos(x)$ and thus

$$\cos(|x|) = \cos(x). \tag{3.2}$$

Then, the cosine of the angular distance between two points can be simplified to

$$\begin{aligned}\cos(\delta_\varphi(\mathcal{P},\mathcal{Q})) &= \cos\big(\pi - |\pi - |\varphi(\mathcal{P}) - \varphi(\mathcal{Q})|\big|\big)\\ &\overset{(3.1)}{=} -\cos\big(|\pi - |\varphi(\mathcal{P}) - \varphi(\mathcal{Q})|\big)\\ &\overset{(3.2)}{=} -\cos\big(\pi - |\varphi(\mathcal{P}) - \varphi(\mathcal{Q})|\big)\\ &\overset{(3.1)}{=} \cos\big(|\varphi(\mathcal{P}) - \varphi(\mathcal{Q})|\big)\\ &\overset{(3.2)}{=} \cos\big(\varphi(\mathcal{P}) - \varphi(\mathcal{Q})\big).\end{aligned}$$

Consequently, the hyperbolic distance between $\mathcal{P}$ and $\mathcal{Q}$ can be written as

$$\begin{aligned}\delta_{\mathbb{H}^2_\zeta}(\mathcal{P},\mathcal{Q}) = \frac{1}{\zeta}\operatorname{acosh}\big(&\cosh(\zeta r(\mathcal{P}))\cosh(\zeta r(\mathcal{Q}))\\ &- \sinh(\zeta r(\mathcal{P}))\sinh(\zeta r(\mathcal{Q}))\cos(\varphi(\mathcal{P}) - \varphi(\mathcal{Q}))\big).\end{aligned}$$

Note that this distance depends on the choice of the curvature ζ, which mainly acts on the radii of the two points. In fact, we can utilize this to scale the distances between $\mathcal{P}$ and $\mathcal{Q}$, by changing the curvature and adjusting the radii accordingly, as formalized in the following lemma.

Lemma 3.1. *Let $\zeta > 0$ be a constant and let $\mathcal{P}, \mathcal{Q} \in \mathbb{H}^2_\zeta$ be two points. Further, consider $\zeta' = \zeta/c$ for a constant $c > 0$. Then, the hyperbolic distance between the points $\mathcal{P}' = (c \cdot r(\mathcal{P}), \varphi(\mathcal{P})) \in \mathbb{H}^2_{\zeta'}$ and $\mathcal{Q}' = (c \cdot r(\mathcal{Q}), \varphi(\mathcal{Q})) \in \mathbb{H}^2_{\zeta'}$ is given by*

$$\delta_{\mathbb{H}^2_{\zeta'}}(\mathcal{P}',\mathcal{Q}') = c \cdot \delta_{\mathbb{H}^2_\zeta}(\mathcal{P},\mathcal{Q}).$$

Proof. We start by computing $\delta_{\mathbb{H}^2_{\zeta'}}(\mathcal{P}',\mathcal{Q}')$ using the distance function as defined above.

$$\begin{aligned}\delta_{\mathbb{H}^2_{\zeta'}}(\mathcal{P}',\mathcal{Q}') = \frac{1}{\zeta'}\operatorname{acosh}\Big(&\cosh\big(\zeta' r(\mathcal{P}')\big)\cosh\big(\zeta' r(\mathcal{Q}')\big)\\ &- \sinh\big(\zeta' r(\mathcal{P}')\big)\sinh\big(\zeta' r(\mathcal{Q}')\big)\cdot\cos\Big(\delta_\varphi(\mathcal{P}',\mathcal{Q}')\Big)\Big).\end{aligned}$$

There, we can now utilize that $r(\mathcal{P}') = c \cdot r(\mathcal{P})$ and $r(\mathcal{Q}') = c \cdot r(\mathcal{Q})$, together

with the fact that $\delta_\varphi(\mathcal{P}', \mathcal{Q}') = \delta_\varphi(\mathcal{P}, \mathcal{Q})$, which yields

$$\delta_{\mathbb{H}^2_{\zeta'}}(\mathcal{P}', \mathcal{Q}') = \frac{1}{\zeta'} \operatorname{acosh}\Big(\cosh\big(\zeta' c \cdot r(\mathcal{P})\big) \cosh\big(\zeta' c \cdot r(\mathcal{Q})\big) - \sinh\big(\zeta' c \cdot r(\mathcal{P})\big) \sinh\big(\zeta' c \cdot r(\mathcal{Q})\big) \cdot \cos\Big(\delta_\varphi(\mathcal{P}, \mathcal{Q})\Big)\Big).$$

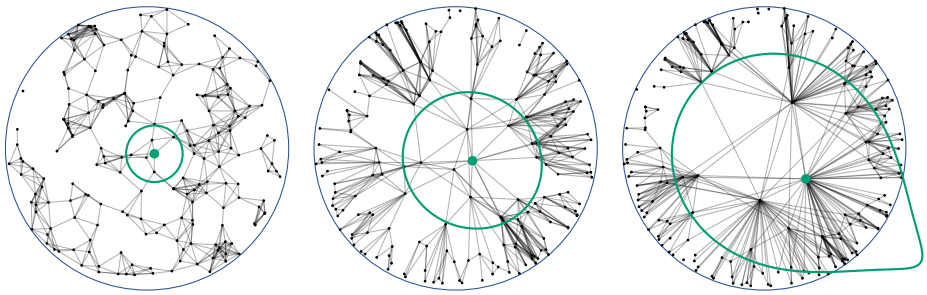
Moreover, we have $\zeta' = \zeta / c$ and thus $\zeta' c = \zeta$, and obtain

$$\begin{aligned}
\delta_{\mathbb{H}^2_{\zeta'}}(\mathcal{P}', \mathcal{Q}') &= \frac{1}{\zeta'} \operatorname{acosh}\Big(\cosh\big(\zeta r(\mathcal{P})\big) \cosh\big(\zeta r(\mathcal{Q})\big) - \sinh\big(\zeta r(\mathcal{P})\big) \sinh\big(\zeta r(\mathcal{Q})\big) \cdot \cos\Big(\delta_\varphi(\mathcal{P}, \mathcal{Q})\Big)\Big) \\
&= \frac{\zeta}{\zeta} \cdot \frac{1}{\zeta'} \operatorname{acosh}\Big(\cosh\big(\zeta r(\mathcal{P})\big) \cosh\big(\zeta r(\mathcal{Q})\big) - \sinh\big(\zeta r(\mathcal{P})\big) \sinh\big(\zeta r(\mathcal{Q})\big) \cdot \cos\Big(\delta_\varphi(\mathcal{P}, \mathcal{Q})\Big)\Big) \\
&= \frac{\zeta}{\zeta'} \cdot \delta_{\mathbb{H}^2_\zeta}(\mathcal{P}, \mathcal{Q}) \\
&= c \cdot \delta_{\mathbb{H}^2_\zeta}(\mathcal{P}, \mathcal{Q}).
\end{aligned}$$

□

In the remainder of this thesis, we follow the commonly used constraint of fixing $K = -1$. That is, we set $\zeta = 1$ and use the shorthand notation $\mathbb{H}^2_1 = \mathbb{H}^2$. For our purposes, this is not actually a constraint since we can simply translate point sets in hyperbolic planes of different curvatures to $\mathbb{H}^2$ and take the scaled distances into account. However, by choosing $\zeta = 1$ the function describing the hyperbolic distance between two point $\mathcal{P}, \mathcal{Q} \in \mathbb{H}^2$ simplifies to

$$\begin{aligned}
\delta_{\mathbb{H}^2}(\mathcal{P}, \mathcal{Q}) &= \operatorname{acosh}\big(\cosh(r(\mathcal{P})) \cosh(r(\mathcal{Q})) - \sinh(r(\mathcal{P})) \sinh(r(\mathcal{Q})) \cos(\delta_\varphi(\mathcal{P}, \mathcal{Q}))\big) \\
&= \operatorname{acosh}\big(\cosh(r(\mathcal{P})) \cosh(r(\mathcal{Q})) - \sinh(r(\mathcal{P})) \sinh(r(\mathcal{Q})) \cos(\varphi(\mathcal{P}) - \varphi(\mathcal{Q}))\big).
\end{aligned} \tag{3.3}$$



$R' = 0.001$, $R = R'/5$ $R' = 4.9$, $R = R'/2$ $R' = 8.5$, $R = R'$

Figure 3.2: Hyperbolic unit disk graphs with different ground space and threshold radii. The representations have been scaled such that the ground spaces *appear* to have the same size, when in fact they are very different, as indicated by the different values for R'. **(Left)** The ground space is very small and the threshold radius even smaller, leading to grid-like structures. **(Center)** Ground space and threshold radius are increased, hierarchies start form but grid like structures remain. **(Right)** Ground space and threshold have the same large value, leading to hierarchical structures.

3.2 Hyperbolic Unit Disk Graphs

Recall that randomly generated Euclidean unit disk graphs are one way to create graphs that exhibit clustering, but that the resulting networks feature a homogeneous degree distribution (see Figure 1.3). The basic idea is to distribute points uniformly at random in a region in the Euclidean plane and to connect each point to all other points that lie within a disk of a fixed radius around it.

In the following, we generalize the class of unit disk graphs by replacing the Euclidean with the hyperbolic plane, which is based on joint work with Thomas Bläsius, Tobias Friedrich, and Daniel Stephan [Blä+23]. Within the resulting class, we define the subclass of *strongly hyperbolic unit disk graphs*, as a counterpart to Euclidean unit disk graphs. Given a graph $G = (V, E)$, a *(Euclidean) unit disk representation* of G is a mapping $p \colon V \to \mathbb{R}^2$ together with a *threshold radius* R such that $\{u, v\} \in E$ if and only if the distance between $p(u)$ and $p(v)$ is at most R. The graph G is a *(Euclidean) unit disk graph* if it has a unit disk representation. The terms *hyperbolic unit disk representation* and *hyperbolic unit disk graph* are defined analogously, except that p maps to the hyperbolic plane $\mathbb{H}^2$ instead of the Euclidean plane $\mathbb{R}^2$. We note that the threshold radius R is part of the representation and can thus depend on the graph. As argued

before, the choice of R does not matter in Euclidean space, since scaling R and all coordinates $p(\cdot)$ by the same factor yields the same adjacencies. However, in the hyperbolic plane $\mathbb{H}^2$ of *fixed* curvature -1, there is no scaling operation that shrinks radii while leaving relative distances intact. The term "unit disk" is still justified as we could instead fix $R = 1$ and allow different curvatures (see Lemma 3.1).

A graph is a *strongly hyperbolic unit disk graph* if it admits a hyperbolic unit disk representation in which p maps all vertices to points within a disk whose radius is equal to the threshold radius R.

3.2.1 The Structure of Unit Disk Graphs

In the following, we discuss how different hyperbolic unit disk representations lead to different structures in hyperbolic unit disk graphs, as illustrated in Figure 3.2. For better understanding, we recommend using the interactive visualization[2] while reading the following. It lets the user change the size of the ground space, allowing to smoothly transition between Euclidean and strongly hyperbolic unit disk graphs. We note that, throughout the thesis, we draw edges of networks in the hyperbolic plane as straight lines and *not* bent towards the origin. This improves the clarity of the drawings and better highlights the structure of the networks. Nevertheless, it is important to keep in mind that distances in the resulting drawings are typically larger than suggested by the drawn edges.

Let $\mathcal{D}_R(\mathcal{P})$ denote a disk of radius R centered at a point $\mathcal{P}$. If $\mathcal{P}$ is the origin of the considered space, we omit the center and simply write $\mathcal{D}_R$ instead. For a radius R', consider a disk $\mathcal{D}_{R'} \subset \mathbb{R}^2$ in the Euclidean plane and assume we distribute vertices evenly in it. Then the Euclidean unit disk graph, obtained by connecting any two points whose Euclidean distance is at most a threshold R, resembles a grid-like structure (with a density depending on R and the radius of R' of the ground space). As the hyperbolic plane resembles the Euclidean plane locally, we can achieve the same grid-like structures by choosing a disk $\mathcal{D}_{R'} \subset \mathbb{H}^2$ with a very small disk radius R', together an even smaller threshold radius R, as shown in Figure 3.2 (left). Beyond that, we can increase the radius of $\mathcal{D}_{R'}$ and the threshold R, and start to observe the formation of hierarchical structures, which are not obtainable with Euclidean unit disks, see Figure 3.2 (center). In

2 https://thobl.github.io/hyperbolic-unit-disk-graph/

the strongly hyperbolic setting (where $R = R'$), we only have hierarchical and no grid-like structures, as shown in Figure 3.2 (right).

Thus, to paint the big picture, hyperbolic unit disk graphs comprise two extremes: Euclidean unit disk graphs with grid-like structures on one side and strongly hyperbolic unit disk graphs with hierarchical structures on the other. Therefore, if we want to design algorithms for grid-like structures, it makes sense to analyze them on Euclidean unit disk graphs. For hierarchical structures, strongly hyperbolic unit disk graphs are a good choice. In particular, this allows us to model graphs that feature a heterogeneous degree distribution, while also exhibiting clustering due to the underlying geometry.

3.2.2 Adjacency in Strongly Hyperbolic Unit Disk Graphs

Recall that a strongly hyperbolic unit disk graph G is equipped with a representation p, mapping the vertices into a disk of radius R that, without loss of generality, is centered at the origin of the hyperbolic plane. From now on and throughout the thesis, we associate a vertex v with its mapping $p(v)$. Moreover, we denote the set of vertices that lie in an area $\mathcal{A}$ with $V(\mathcal{A})$.

By definition, two vertices in the graph G are adjacent, if and only if their hyperbolic distance is at most R. Consequently, we can imagine that each vertex v is equipped with a neighborhood disk $\mathcal{D}_R(v)$ containing all its neighbors. That is, $N(v) = V(\mathcal{D}_R(v))$. The following lemma shows that moving such a neighborhood disk closer to the center of $\mathcal{D}_R$ only increases the region of $\mathcal{D}_R$ that it covers. From this, we can derive that moving a vertex towards the center of $\mathcal{D}_R$ only increases its neighborhood.

Lemma 3.2. *Let R be a radius and let $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{D}_R$ be points with $r(\mathcal{P}_1) \leq r(\mathcal{P}_2)$ and $\varphi(\mathcal{P}_1) = \varphi(\mathcal{P}_2)$. Then, $\mathcal{D}_R(\mathcal{P}_1) \supseteq \mathcal{D}_R(\mathcal{P}_2) \cap \mathcal{D}_R$.*

Proof. Let $\mathcal{P} \in \mathcal{D}_R(\mathcal{P}_2) \cap \mathcal{D}_R$ be a point and note that $\delta_{\mathbb{H}^2}(\mathcal{P}, \mathcal{P}_2) \leq R$. Now consider the triangle spanned by the points $\mathcal{P}, \mathcal{P}_2$, and the origin O. This triangle is completely contained in the disk $\mathcal{D}_R(\mathcal{P})$, as $\delta_{\mathbb{H}^2}(\mathcal{P}, \mathcal{P}_2) \leq R$ and $r(\mathcal{P}) \leq R$, as shown in Figure 3.3. Since disks are convex and $\mathcal{P}_1$ lies on the line from O to $\mathcal{P}_2$, it is part of the triangle and therefore also contained in the disk. Consequently, $\delta_{\mathbb{H}^2}(\mathcal{P}, \mathcal{P}_1) \leq R$ and thus $\mathcal{P} \in \mathcal{D}_R(\mathcal{P}_1)$. □

Corollary 3.3. *Let G be a strongly hyperbolic unit disk graph with radius R and let v_1, v_2 be vertices with $r(v_1) \leq r(v_2) \leq R$ and $\varphi(v_1) = \varphi(v_2)$. Then, $N(v_1) \supseteq N(v_2)$.*

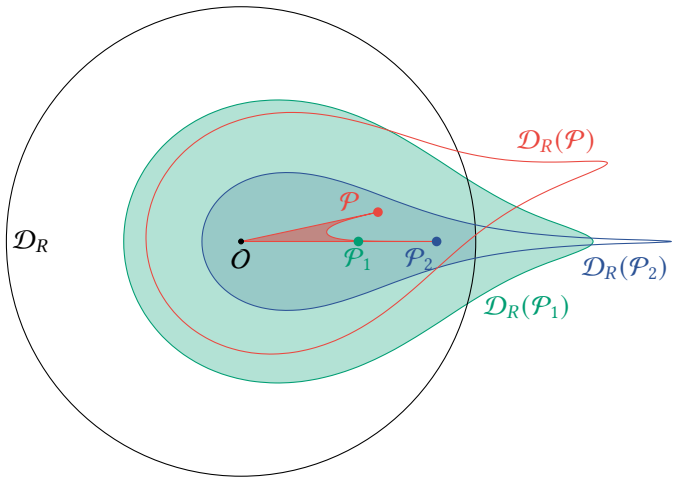


Figure 3.3: Visualization of the proof of Lemma 3.2. Point $\mathcal{P}_1$ has a smaller radius than $\mathcal{P}_2$, both having the same angular coordinate. Consequently, $\mathcal{D}_R(\mathcal{P}_1)$ (green region) is a superset of $\mathcal{D}_R(\mathcal{P}_2) \cap \mathcal{D}_R$ (blue region). The triangle formed by the points $\mathcal{P}, \mathcal{P}_\in$, and O is contained in $\mathcal{D}_R(\mathcal{P})$ (both red).

In the following, we investigate in greater detail under which circumstances two vertices are adjacent. Consider two vertices v_1 and v_2 in G with radii r_1 and r_2, respectively. The two are adjacent, if their hyperbolic distance is at most R. Clearly, this is the case, if $r_1 + r_2 \leq R$. However, when $r_1 + r_2 > R$, it depends on their angular distance $\delta_\varphi(v_1, v_2)$ whether their hyperbolic distance is at most R. Considering the hyperbolic distance function in Equation (3.3), it is easy to see that this distance increases with increasing angular distance $\delta_\varphi(v_1, v_2) \in [0, \pi]$, since $\cos(x)$ decreases for increasing $x \in [0, \pi]$ and $\operatorname{acosh}(x)$ increases in x. We use $\theta(r_1, r_2)$ to denote the angular distance, such that the hyperbolic distance between v_1 and v_2 is exactly R. Then, for $\delta_\varphi(v_1, v_2) \leq \theta(r_1, r_2)$ we have $\delta_{\mathbb{H}^2}(v_1, v_2) \leq R$, meaning v_1 and v_2 are adjacent. Whereas for $\delta_\varphi(r_1, r_2) > \theta(r_1, r_2)$ the two vertices are not adjacent. We can compute $\theta(r_1, r_2)$ by using the hyperbolic distance function in Equation (3.3), setting the distance equal to R, and solving for the angular distance. That is,

$$\theta(r_1, r_2) = \operatorname{acos}\left(\frac{\cosh(r_1)\cosh(r_2) - \cosh(R)}{\sinh(r_1)\sinh(r_2)}\right). \tag{3.4}$$

The following lemma, gives upper and lower bounds on $\theta(r_1, r_2)$ that are easier to work with than this expression.

Lemma 3.4. *Let $R > 0$ and $r_1, r_2 \in (0, R]$ with $r_1 + r_2 \geq R$ be given. Then,*

$$2\sqrt{e^{R-r_1-r_2} + e^{-R-r_1-r_2} - (e^{-2r_1} + e^{-2r_2})} \leq \theta(r_1, r_2) \leq \pi\sqrt{e^{R-r_1-r_2}}.$$

Proof. We start by applying the cosine function on both sides of Equation (3.4), which makes it easier to deal with the right hand side for now. This yields

$$\cos(\theta(r_1, r_2)) = \frac{\cosh(r_1)\cosh(r_2) - \cosh(R)}{\sinh(r_1)\sinh(r_2)}. \tag{3.5}$$

We consider the upper bound on $\theta(r_1, r_2)$ first. Note that we aim to eventually apply the inverse cosine function to revert the above step. Since this function is monotonically decreasing, we first determine a *lower* bound on $\cos(\theta(r_1, r_2))$, in order to obtain an upper bound on $\theta(r_1, r_2)$. Recall that $\cosh(x) = (e^x + e^{-x})/2$ and $\sinh(x) = (e^x - e^{-x})/2$, and note that $\sinh(x) \leq e^x/2$. Thus, the above equation can be bounded by

$$\begin{aligned}
\cos(\theta(r_1, r_2)) &\geq \frac{1/4(e^{r_1} + e^{-r_1})(e^{r_2} + e^{-r_2}) - 1/2(e^R + e^{-R})}{1/4e^{r_1+r_2}} \\
&= \frac{e^{r_1+r_2} + e^{r_1-r_2} + e^{r_2-r_1} + e^{-r_1-r_2} - 2e^R - 2e^{-R}}{e^{r_1+r_2}} \\
&= 1 - 2e^{R-r_1-r_2} + e^{-2r_1} + e^{-2r_2} + e^{-2(r_1+r_2)} - 2e^{-R-r_1-r_2}.
\end{aligned}$$

We now argue that the remaining expression can be bounded by dropping the last four terms since their sum is non-negative. First note that $e^x \geq 0$ for all $x \in \mathbb{R}$. Consequently, the second to last term is non-negative and it remains to show that $e^{-2r_1} + e^{-2r_2} \geq 2e^{-R-r_1-r_2}$, which can be done by showing that $e^{-2r_1}, e^{-2r_2} \geq e^{-R-r_1-r_2}$. In the following, we show that this is the case for e^{-2r_1}. The proof for e^{-2r_2} is analogous. Note that $r_1 - r_2 \leq R$, since $r_1, r_2 \in (0, R]$ by assumption. It follows that $r_1 \leq R + r_2$ and thus $e^{-2r_1} \geq e^{-R-r_1-r_2}$. We can conclude that $\cos(\theta(r_1, r_2)) \geq 1 - 2e^{R-r_1-r_2}$. The claimed upper bound now follows by applying the inverse cosine and observing that $\operatorname{acos}(1 - x) \leq \pi\sqrt{x/2}$ holds for all $x \in [0, 2]$.

It remains to prove that the claimed lower bound on $\theta(r_1, r_2)$ is valid. Again, we start with Equation (3.5). However, this time we determine an *upper* bound

on $\cos(\theta(r_1, r_2))$. First, we apply the identity

$$\cosh(x)\cosh(y) = \sinh(x)\sinh(y) + \cosh(x-y),$$

which yields

$$\begin{aligned}\cos(\theta(r_1, r_2)) &= \frac{\sinh(r_1)\sinh(r_2) + \cosh(r_1 - r_2) - \cosh(R)}{\sinh(r_1)\sinh(r_2)} \\ &= 1 - \frac{\cosh(R) - \cosh(r_1 - r_2)}{\sinh(r_1)\sinh(r_2)}.\end{aligned}$$

Using the definition of cosh and the fact that $\sinh(x) \leq e^x/2$, we conclude that

$$\begin{aligned}\cos(\theta(r_1, r_2)) &\leq 1 - \frac{1/2(e^R + e^{-R}) - 1/2(e^{r_1 - r_2} + e^{r_2 - r_1})}{1/4e^{r_1 + r_2}} \\ &= 1 - 2(e^{R - r_1 - r_2} + e^{-R - r_1 - r_2} - (e^{-2r_2} + e^{-2r_1})).\end{aligned}$$

The claim then follows by applying the inverse cosine function and observing that $\operatorname{acos}(1-x) \geq \sqrt{2x}$ is valid for all $x \in [0, 2]$. □

We note that, while the above bounds are easier to work with than the exact function and are generally applicable due to the few constraints on the considered radii, the lower bound is still a bit tedious to work with. However, by introducing some minor requirements, we can obtain a slightly weaker bound that can be worked with more easily.

Corollary 3.5. *Let $R \geq 1$ and $r_1, r_2 \in (0, R]$ with $r_1 + r_2 \geq R$ and $|r_1 - r_2| \leq R - 1$ be given. Then,*

$$\sqrt{e^{R - r_1 - r_2}} \leq \theta(r_1, r_2) \leq \pi\sqrt{e^{R - r_1 - r_2}}.$$

Proof. The upper bound immediately follows from Lemma 3.4. By utilizing the lower bound from the same lemma, we obtain

$$\begin{aligned}\theta(r_1, r_2) &\geq 2\sqrt{e^{R - r_1 - r_2} + e^{-R - r_1 - r_2} - (e^{-2r_1} + e^{-2r_2})}. \\ &\geq 2\sqrt{e^{R - r_1 - r_2} - (e^{-2r_1} + e^{-2r_2})}. \\ &= 2\sqrt{e^{R - r_1 - r_2}\left(1 - e^{-R}(e^{r_1 - r_2} + e^{-(r_1 - r_2)})\right)},\end{aligned}$$

where the second inequality is valid since $e^{-R - r_1 - r_2} \geq 0$. To prove the claim, it

thus suffices to show that the remaining negative part is at most 3/4, which can be done as follows. First note that

$$e^{-R}(e^{r_1-r_2}+e^{-(r_1-r_2)}) = 2e^{-R}\cdot\frac{1}{2}(e^{r_1-r_2}+e^{-(r_1-r_2)}) = 2e^{-R}\cdot\cosh(r_1-r_2).$$

Now note that $\cosh(x)$ is symmetric about the y-axis and thus $\cosh(r_1-r_2) = \cosh(|r_1-r_2|)$. Moreover, since $\cosh(x)$ is monotonically increasing for $x \geq 0$, we can utilize the assumption that $|r_1-r_2| \leq R-1$ to conclude

$$e^{-R}(e^{r_1-r_2}+e^{-(r_1-r_2)}) \leq 2e^{-R}\cdot\cosh(R-1).$$

Finally, since $\cosh(x) = 1/2(e^x+e^{-x}) \leq e^x$ for all $x \geq 0$, we obtain

$$e^{-R}(e^{r_1-r_2}+e^{-(r_1-r_2)}) \leq 2e^{-R}\cdot e^{R-1} = 2/e \leq 3/4. \qquad \square$$

Apart from the above bounds, we highlight another property of the function $\theta(r_1,r_2)$, for the special case where $r_1 = r_2$.

Lemma 3.6. *The function $\theta(r,r)$ is monotonically decreasing for $r \geq 0$.*

Proof. Consider the definition of $\theta(r_1,r_2)$ in Equation (3.4). By utilizing the fact that $r_1 = r_2 = r$, the equation simplifies to

$$\theta(r,r) = \operatorname{acos}\left(\frac{\cosh(r)^2-\cosh(r)}{\sinh(r)^2}\right).$$

We can now apply the identities $\cosh(x)^2 = (\cosh(2x)+1)/2$ and $\sinh(x)^2 = (\cosh(2x)-1)/2$, both being valid for $x \in \mathbb{R}$, to obtain

$$\begin{aligned}
\theta(r,r) &= \operatorname{acos}\left(\frac{1/2(\cosh(2r)+1)-\cosh(r)}{1/2(\cosh(2r)-1)}\right)\\
&= \operatorname{acos}\left(\frac{(\cosh(2r)+1)-2\cosh(r)}{\cosh(2r)-1}\right)\\
&= \operatorname{acos}\left(\frac{\cosh(2r)-1+2-2\cosh(r)}{\cosh(2r)-1}\right)\\
&= \operatorname{acos}\left(1-2\frac{\cosh(r)-1}{\cosh(2r)-1}\right).
\end{aligned}$$

Further, utilizing the fact that

$$\frac{\cosh(x)-1}{\cosh(2x)-1} = \frac{1}{2\cosh(x)+2},$$

which is valid for all $x \in \mathbb{R}$, the above term can be simplified to

$$\theta(r,r) = \operatorname{acos}\left(1 - \frac{1}{\cosh(r)+1}\right).$$

Note that $\cosh(x)$ is monotonically increasing for $x \geq 0$, and so is the argument in the inverted cosine. The claim follows as acos is monotonically decreasing. □

3.2.3 Cliques in Strongly Hyperbolic Unit Disk Graphs

As mentioned above, strongly hyperbolic unit disk graphs exhibit clustering due to the underlying geometry. In the following, we examine how this affects the formation of cliques. We start by showing that the vertices lying in a disk $\mathcal{D}_R(\mathcal{P})$ having smaller radius than $\mathcal{P}$, form two cliques. More precisely, we say that a vertex set $S \subseteq V$ can be *covered by k cliques*, if there exists a partitioning $S_1, \ldots, S_k$ of S such that the induced subgraphs $G[S_i]$ for $i \in [k]$ are cliques.

Lemma 3.7. *Let G be a strongly hyperbolic unit disk graph with radius $R > 0$ and let $\mathcal{P} \in \mathcal{D}_R$ be a point with $r(\mathcal{P}) = r$. Then, $V(\mathcal{D}_R(\mathcal{P}) \cap \mathcal{D}_r)$ can be covered by two cliques.*

Proof. Assume without loss of generality that $\varphi(\mathcal{P}) = 0$. We divide the region $\mathcal{D}_R(\mathcal{P}) \cap \mathcal{D}_r$ into two halves $\mathcal{A}$ and $\mathcal{A}'$ containing all points with angles in $[0, \pi)$ and $[\pi, 2\pi)$, respectively, as illustrated in Figure 3.4. The goal now is to show that the vertices in $V(\mathcal{A})$ and the ones $V(\mathcal{A}')$ induce a clique. More precisely, we show that this is the case for $\mathcal{A}$. For symmetry reasons this then also holds for $\mathcal{A}'$. Consider two vertices $v_1, v_2 \in \mathcal{A}$ and assume without loss of generality that $\varphi(v_1) \leq \varphi(v_2)$. Since $v_2 \in \mathcal{A} \subseteq \mathcal{D}_R(\mathcal{P})$ and since by Lemma 3.2 moving $\mathcal{D}_R(\mathcal{P})$ towards the origin increases the region of $\mathcal{D}_R$ that it covers, we know that v_2 is contained in the disk $\mathcal{D}_R(\mathcal{P}')$ for $\mathcal{P}' = (r(v_1), 0)$ (dark green in Figure 3.4). It follows that $\delta_{\mathbb{H}^2}(\mathcal{P}', v_2) \leq R$. Note that v_1 has the same radius as $\mathcal{P}'$ and that $\delta_\varphi(\mathcal{P}', v_2) \geq \delta_\varphi(v_1, v_2)$. As established above, decreasing the angular distance between two points with fixed radii decreases their hyperbolic distance. Therefore, $\delta_{\mathbb{H}^2}(v_1, v_2) \leq \delta_{\mathbb{H}^2}(\mathcal{P}', v_2) \leq R$, meaning v_1 and v_2 are adjacent. □

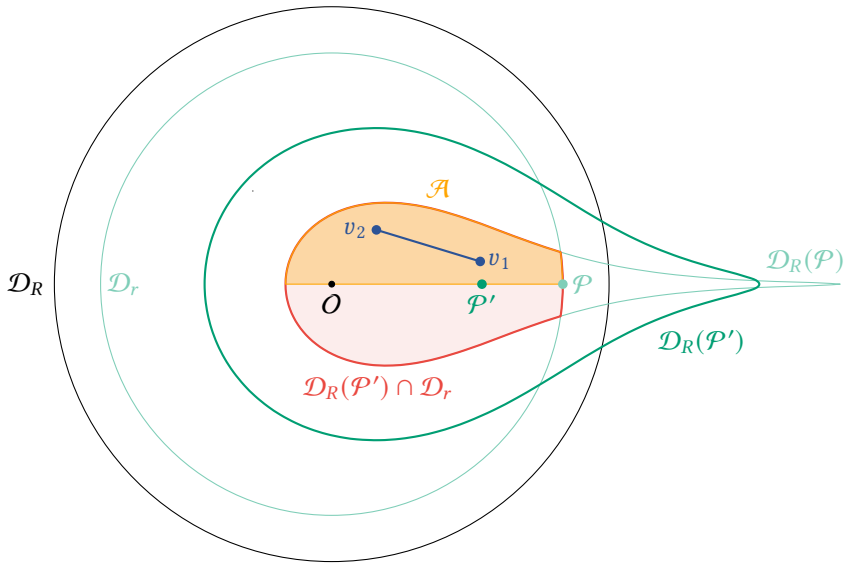


Figure 3.4: Visualization of the proof of Lemma 3.7. Vertices v_1, v_2 (blue) are in the half $\mathcal{A}$ (orange) of the region $\mathcal{D}_R(\mathcal{P}) \cap \mathcal{D}_r$ (red) and are adjacent.

We note that the above lemma implies that the neighbors of a vertex v with smaller radius than v form two cliques. We continue by investigating the number of cliques required to cover a strongly hyperbolic unit disk graph.

Lemma 3.8. *Let G be a strongly hyperbolic unit disk graph with radius $R > 0$. Then, G can be covered by $\max\{2\pi\sqrt{2}, 2\pi e^{R/2}\}$ cliques.*

Proof. To prove the claim, we utilize the underlying geometry by covering the ground space $\mathcal{D}_R$ with a set of k disks $\mathcal{D}^1, \dots, \mathcal{D}^k$, such that each $V(\mathcal{D}^i)$ for $i \in [k]$ can be covered by two cliques. All of these disks have radius R and their centers lie on the boundary of the disk $\mathcal{D}_R$. The center of the first disk has an angular coordinate of 0. All other disks $\mathcal{D}^i$ are placed at an angular distance of $2\theta(R, R)$ to their predecessor $\mathcal{D}^{i-1}$ in counterclockwise direction. See Figure 3.5 for an illustration. As a consequence, the boundaries of two consecutive disks intersect on the boundary of $\mathcal{D}_R$, which is therefore covered completely by the k disks. It follows that each vertex is contained in at least one disk $\mathcal{D}^i$.

Since by Lemma 3.7 each $V(\mathcal{D}^i)$ for $i \in [k]$ can be covered by two cliques,

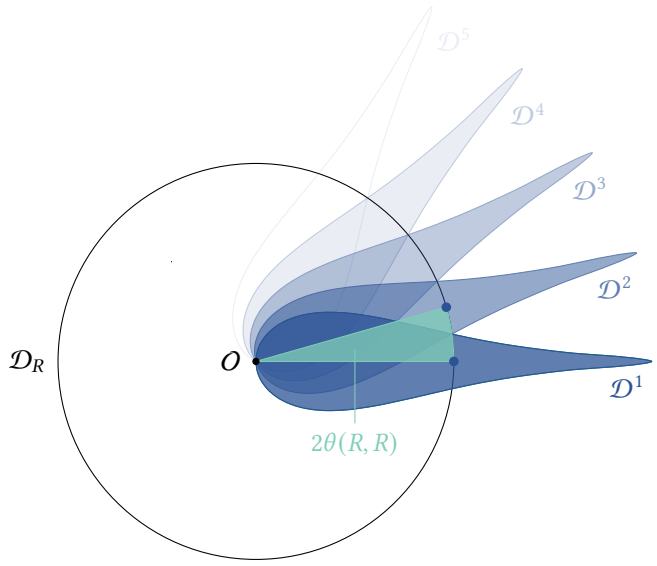


Figure 3.5: Visualization of the proof of Lemma 3.8, showing the first five of the k disks $\mathcal{D}^1, \dots, \mathcal{D}^k$ (blue) and the angular distance between two consecutive centers (green).

it suffices to show that $k \leq \max\{\pi\sqrt{2}, \pi e^{R/2}\}$ in order to finish the proof. To this end, recall that two consecutive disks are placed at an angular distance of $2\theta(R,R)$. Consequently, it takes $k = 2\pi/(2\theta(R,R)) = \pi/\theta(R,R)$ disks to cover the whole disk $\mathcal{D}_R$. Using Lemma 3.4 we can conclude

$$\begin{aligned}\theta(R,R) &\geq 2\sqrt{e^{-R} + e^{-3R} - 2e^{-2R}} \\ &= 2\sqrt{\left(e^{-R/2} - e^{-3/2\cdot R}\right)^2} \\ &= 2\left(e^{-R/2} - e^{-3/2\cdot R}\right) \\ &= 2e^{-R/2}(1 - e^{-R}).\end{aligned}$$

It follows that k can be bounded by

$$k = \frac{\pi}{\theta(R,R)} \leq \frac{\pi}{2e^{-R/2}\left(1 - e^{-R}\right)} = \pi e^{R/2} \cdot \frac{1}{2\left(1 - e^{-R}\right)}. \tag{3.6}$$

We now distinguish between two cases depending on the size of R and start

with $R < \log(2)$. Recall that the function $\theta(R, R)$ is monotonically decreasing in R (see Lemma 3.6). As a consequence, we have $\theta(R, R) \geq \theta(\log(2), \log(2))$. Then, it follows that

$$\begin{aligned} k &\leq \frac{\pi}{\theta(\log(2), \log(2))} \\ &\leq \pi e^{\log(2)/2} \frac{1}{2\left(1 - e^{-\log(2)}\right)} \\ &= \pi\sqrt{2}, \end{aligned}$$

which we account for with the first part of the maximum. When $R \geq \log(2)$, note that we have $(1 - e^{-R}) \geq 1/2$. Consequently, we can bound the last fraction in Equation (3.6) by 1, which yields the claim. □

3.2.4 Related Concepts

To the best of our knowledge, intersection graphs of hyperbolic unit disks, or hyperbolic unit balls, have so far only been considered by Kisfaludi-Bak [Kis20]. There, for every $\rho > 0$, a graph is said to be in the graph class $\mathrm{UBG}_{\mathbb{H}^d}(\rho)$ (*UBG = unit ball graph*) if its vertices can be mapped into $\mathbb{H}^d$ such that vertices have distance at most 2ρ if and only if they are adjacent. There are two core differences compared to our definition of hyperbolic unit disk graphs. First, it allows for higher dimensions. Secondly, it is parameterized by the radius, i.e., $\mathrm{UBG}_{\mathbb{H}^d}(\rho)$ describes an infinite family of graph classes rather than a single class.

This second difference is somewhat subtle but rather important. Consider the class $\mathrm{UBG}_{\mathbb{H}^d}(\rho)$ for a fixed radius ρ. Moreover, assume we want to study graphs in $\mathrm{UBG}_{\mathbb{H}^d}(\rho)$ that are sparse; for the sake of argument, assume constant average degree. Then, for an increasing number of vertices n, the region of $\mathbb{H}^d$ spanned by the vertices has to grow, as otherwise the density of the graph grows with n. Thus, for sufficiently large n, the radius ρ is arbitrarily small compared to the region spanned by the vertices, yielding grid-like structures (see discussion above). Consequently, for fixed ρ, large graphs in $\mathrm{UBG}_{\mathbb{H}^d}(\rho)$ are grid-like rather than hierarchical. This means that asymptotic statements for the classes $\mathrm{UBG}_{\mathbb{H}^d}(\rho)$ do not translate to the hierarchical structures in the class of strongly hyperbolic unit disk graphs.

A second related concept are *hyperbolic random graphs*, which we consider in greater detail in the following section.

3.3 Hyperbolic Random Graphs

Hyperbolic random graphs were introduced by Krioukov et al., as a model that features a power-law degree distribution and high clustering [Kri+10]. They are basically *random* strongly hyperbolic unit disk graphs[3]. In particular, every hyperbolic random graph is a strongly hyperbolic unit disk graph and thus any statement shown for the latter also holds for the former.

3.3.1 Definition

Formally, a hyperbolic random graph with n vertices is obtained by first assigning each vertex a random point in a disk $\mathcal{D}_R \subset \mathbb{H}^2$ of radius R that is centered at the origin. Afterwards, two vertices u and v are connected by an edge, if and only if $\delta_{\mathbb{H}^2}(u,v) \leq R$. The choice of R affects the properties of the generated network. In particular, to obtain a power-law exponent $\beta = 2\alpha + 1$ (for $\alpha \in (1/2, 1)$) with high probability and an expected average degree of κ, both of which are assumed to be constant, R has to be chosen as $R = 2\log(n) + C$ for a constant

$$C = 2\log\left(\frac{2}{\pi\kappa}\left(\frac{\alpha}{\alpha - 1/2}\right)^2 (1 + o(1))\right). \qquad \text{([GPP12, Theorems 3 \& 4])}$$

The coordinates of the vertices are independently drawn as follows. For a vertex the angular coordinate is drawn uniformly at random from $[0, 2\pi)$ and its radius is sampled according to the probability density function

$$f(r) = \frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R) - 1},$$

for $r \in [0, R]$. For $r > R$, $f(r) = 0$. This function can be closely approximated as

$$f(r) = e^{-\alpha(R-r)}\left(1 + \Theta\left(e^{-\alpha R} - e^{-2\alpha r}\right)\right). \qquad \text{([Kro16, Lemma 3.1])}$$

Thus, the joint probability density function of radius and angle is given by

$$f(r,\varphi) = \frac{1}{2\pi}\frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R) - 1} = \frac{\alpha}{2\pi} e^{-\alpha(R-r)}\left(1 + \Theta\left(e^{-\alpha R} - e^{-2\alpha r}\right)\right). \qquad (3.7)$$

3 In general, hyperbolic random graphs feature an additional *temperature* that is not captured by strongly hyperbolic unit disk graphs. Here, we focus on the version without temperature.

3.3.2 Properties

The above probability density function is a natural choice as the probability for a vertex ending up in a certain region is proportional to its area (at least for $\alpha = 1$). Note that the exponential growth in r reflects the fact that the area of a disk grows exponentially with its radius. It follows that most vertices lie close to the boundary of the hyperbolic disk $\mathcal{D}_R$ and only few vertices are close to its center. Moreover, the exponential expansion of space has another affect. As for Euclidean unit disk graphs, we can imagine that each vertex v is equipped with a *neighborhood* disk $\mathcal{D}_R(v) \subset \mathbb{H}^2$, i.e., a disk of radius R centered at v, that contains all other vertices that v is adjacent to. Now if v lies at the center of the ground space $\mathcal{D}_R$ then it covers the hole disk $\mathcal{D}_R$ and is therefore adjacent to all other vertices. However, if v lies on the boundary of the ground space, then its neighborhood disk only covers a $\Theta(1/n)$-portion of the disk, meaning its expected number of neighbors is constant and not linear. It follows that a hyperbolic random graph has few vertices with high degree close to the center of the disk and many vertices with low degree near its boundary, leading to a heterogeneous degree distribution. As mentioned above, this degree distribution follows a power law with exponent $\beta = 2\alpha + 1$, with high probability. The parameter α is assumed to be in the range $(1/2, 1)$, yielding power-law exponents $\beta \in (2, 3)$. Exponents outside of this range are atypical for hyperbolic random graphs. For $\alpha < 1/2$ the average degree of the generated networks diverges and the graph is connected asymptotically almost surely [BFM16]. For $\alpha > 1$ the graphs decompose into small components (of size sublinear in n) [BFM15], the diameter is at most logarithmic [FK18], and the variance of the degree distribution is no longer unbounded.

In this thesis, we focus on the typical range $\alpha \in (1/2, 1)$. There, the variance of the degree sequence is unbounded, leading to very heterogeneous degree distributions. The properties of hyperbolic random graphs in this range are researched comprehensively. If not stated otherwise, the following results hold with high probability. The average degree is constant [BKL19] (its value being κ in expectation), while the maximum degree is bounded by $O(n^{1/(2\alpha-\varepsilon)})$ for $\varepsilon > 0$ [Keu18], with a bound of $n^{1/(2\alpha)+o(1)}$ holding asymptotically almost surely [GPP12]. The obtained networks have a giant component of size $\Omega(n)$ [BKL19]. Prior insights about the giant component give bounds depending on α that hold asymptotically almost surely [BFM13]. All non-giant components have size at most $O(\log(n)^{1/(1-\alpha)})$ [FK18, Corollary 13], with a matching lower bound that

holds asymptotically almost surely [KM19]. The diameter of a hyperbolic random graph is bounded by $\Omega(\log(n))$ from below and by $O(\log(n)^{1/(1-\alpha)})$ from above [FK18], with a logarithmic upper bound that holds asymptotically almost surely [MS19]. Additionally, the average path length in the giant component is $\Theta(\log\log(n))$ asymptotically almost surely [BKL19]. The clustering coefficient is constant (and thus non-vanishing) [BKL19]. The actual constant has been determined as a limit in n [Fou+21]. Moreover, other clustering properties have been researched [CF16b; Ste20], and further insights about the connectivity have been obtained by studying the spectrum of the normalized Laplacian [KM18].

Aside from structural properties, hyperbolic random graphs have also been analyzed from an algorithmic perspective. They can be generated in expected linear time [BKL19; Blä+19]. The maximum clique (subgraph where all pairs of vertices are adjacent) can be computed in polynomial time [BFK18] and there are several algorithmic results based on the fact that hyperbolic random graphs have small balanced separators and therefore sublinear treewidth (they can be decomposed into specific trees whose vertices represent sets of vertices in the original graph) [BFK16]. There is a compression algorithm that can store a hyperbolic random graph using $O(n)$ bits in expectation [BKL19], and a close approximation of the shortest path between two vertices can be found using greedy routing, which visits only $O(\log\log n)$ vertices for most start–destination pairs [Bri+17]. Moreover, bootstrap percolation processes have been studied on hyperbolic random graphs, where it was shown that even small infection rates are sufficient to infect at least a constant fraction of all vertices [CF16a].

In the following, we present some preliminary mathematical tools that are useful for further exploration of algorithmic properties of hyperbolic random graphs in the remainder of the thesis.

3.3.3 Vertex Distribution

Recall that a hyperbolic random graph is obtained by distributing n points at random in the disk $\mathcal{D}_R \subset \mathbb{H}^2$, according to the probability density function $f(r,\varphi)$ in Equation (3.7). The probability that a sampled vertex falls into a given subset $\mathcal{A} \subseteq \mathcal{D}_R$ of the disk is given by its *probability measure* $\mu(\mathcal{A}) = \iint_{\mathcal{A}} f(r,\varphi)\,\mathrm{d}\varphi\,\mathrm{d}r$, which can be thought of as the area of $\mathcal{A}$. Now consider the random variables $X_1,\dots,X_n$ with $X_i = 1$ if vertex i lies in $\mathcal{A}$, i.e., if $i \in V(\mathcal{A})$, and $X_i = 0$ otherwise. Then, the number of vertices in $\mathcal{A}$ is given by $|V(\mathcal{A})| = \sum_{i=1}^{n} X_i$. By the linearity

of expectation, we obtain that the expected number of vertices in $\mathcal{A}$ is

$$\mathbb{E}\big[|V(\mathcal{A})|\big] = \sum_{i=1}^{n} \mathbb{E}[X_i] = n\mu(\mathcal{A}).$$

When analyzing hyperbolic random graphs, there are two types of regions $\mathcal{A}$ that we encounter regularly: disks $\mathcal{D}_r$ with radius r centered at the origin and disks $\mathcal{D}_R(\mathcal{P})$ of radius R centered at a point $\mathcal{P} \in \mathbb{H}^2$. Gugelmann et al. [GPP12, Lemma 3.2] showed that

$$\mu(\mathcal{D}_r) = e^{-\alpha(R-r)}(1 + o(1)), \text{ and} \tag{3.8}$$

$$\mu(\mathcal{D}_R(\mathcal{P})) = \frac{2\alpha e^{-r(\mathcal{P})/2}}{(\alpha - 1/2)\pi}\left(1 \pm O\left(e^{-(\alpha-1/2)r(\mathcal{P})} + e^{-r(\mathcal{P})}\right)\right) = \Theta\left(e^{-r(\mathcal{P})/2}\right). \tag{3.9}$$

Note that, for a vertex v, the measure of $\mathcal{D}_R(v)$ gives the probability that a given vertex lies in the neighborhood of v. Consequently, the expected degree of v is given by

$$\mathbb{E}\big[\deg(v)\big] = \mathbb{E}\big[|V(\mathcal{D}_R(v))|\big] = n\mu(\mathcal{D}_R(v)) = \Theta\left(ne^{-r(v)/2}\right).$$

Moreover, if the expected degree of a vertex v is at least logarithmic, we can apply Corollaries 2.3 and 2.4 to conclude that the actual degree of v matches the expectation up to constant factors, with high probability.

3.3.4 Hyperbolic Random Graphs with n Vertices in Expectation

Even though the vertices are distributed independently from each other, computing the probability for single vertex v to lie in a region $\mathcal{A}$ becomes significantly harder once the positions of some vertices are already known, since that introduces stochastic dependencies. For example, if all n vertices are sampled into $\mathcal{A}$, the probability for a vertex to lie outside of $\mathcal{A}$ is 0. In order to overcome such issues, we use an approach that was already used on hyperbolic random graphs before [BFM15; FK18; FM18; Fou+21; KM15], where the vertex positions in the hyperbolic disk are sampled using an *inhomogeneous Poisson point process*. For a given number of vertices n, we refer to the resulting model as *hyperbolic random graphs with n vertices in expectation*. After analyzing properties of this simpler model, we can translate the results back to the original model, by conditioning

on the fact that the resulting distribution is equivalent to the one originally used for hyperbolic random graphs. More formally, this can be done as follows.

A hyperbolic random graph with n vertices in expectation is obtained using an inhomogeneous Poisson point process to distribute the vertices in the hyperbolic disk. In order to get n vertices in expectation, the corresponding intensity function $f_P(r, \varphi)$ at a point $(r, \varphi) \in \mathcal{D}_R$ is chosen as

$$f_P(r, \varphi) = e^{(R-C)/2} f(r, \varphi),$$

where $f(r, \varphi)$ is the original probability density function used to sample hyperbolic random graphs (see Equation (3.7)). Let P denote the set of random variables representing the points produced by this process. Then P has two properties. First, the number of vertices in P that are sampled into two disjoint areas are independent random variables. Second, the expected number of points in P that fall within an area $\mathcal{A}$ is given by

$$\iint_{\mathcal{A}} f_P(r, \varphi) \, \mathrm{d}\varphi \, \mathrm{d}r = n \iint_{\mathcal{A}} f(r, \varphi) \, \mathrm{d}\varphi \, \mathrm{d}r = n\mu(\mathcal{A}).$$

By the choice of f_P the number of vertices sampled into the disk matches n only in expectation, i.e., $\mathbb{E}\big[|P|\big] = n$. However, we can now recover the original distribution of the vertices, by conditioning on the fact that $|P| = n$, as shown in the following lemma. Intuitively, it states that probabilistic statements on hyperbolic random graphs with n vertices in expectation can be translated to the original hyperbolic random graph model by taking a small penalty in certainty. We note that proofs of how to bound this penalty have been sketched before [FK18; KM15]. For the sake of completeness, we give an explicit proof. In the following, we use G_P to denote a hyperbolic random graph with n vertices in expectation and point set P. Moreover, we use $\boldsymbol{P}$ to denote a property of a graph and for a given graph G we denote the event that G has property $\boldsymbol{P}$ with $E(G, \boldsymbol{P})$.

Lemma 3.9. *Consider a hyperbolic random graph G_P with n vertices in expectation, a property $\boldsymbol{P}$, and a constant $c > 0$, such that $\Pr\big[E(G_P, \boldsymbol{P})\big] = \mathrm{O}\big(|P|^{-c}\big)$. Then, for a hyperbolic random graph G' with n vertices, $\Pr\big[E(G', \boldsymbol{P})\big] = \mathrm{O}\big(n^{-c+1/2}\big)$.*

Proof. The probability that G' has property $\boldsymbol{P}$ can be obtained by taking the probability that a hyperbolic random graph G_P with n vertices in expectation has it, and conditioning on the fact that *exactly* n vertices are produced during

its sampling process. That is,

$$\Pr\left[E(G', \boldsymbol{P})\right] = \Pr\left[E(G_P, \boldsymbol{P}) \,\middle|\, |P| = n\right].$$

This probability can now be computed using the definition of conditional probabilities (see Section 2.2). That is,

$$\Pr\left[E(G_P, \boldsymbol{P}) \,\middle|\, |P| = n\right] = \frac{\Pr\left[E(G_P, \boldsymbol{P}) \cap |P| = n\right]}{\Pr\left[|P| = n\right]}.$$

For the numerator, we have $\Pr\left[E(G_P, \boldsymbol{P})\right] = \mathrm{O}\left(|P|^{-c}\right)$. Constrained to events where $|P| = n$, this yields $\Pr\left[E(G_P, \boldsymbol{P}) \cap |P| = n\right] = \mathrm{O}(n^{-c})$. For the denominator, recall that $|P|$ is a random variable that follows a Poisson distribution with mean n. Therefore, we have

$$\Pr\left[|P| = n\right] = \frac{e^{-n}n^n}{n!} = \Theta\left(n^{-1/2}\right).$$

The quotient can, thus, be bounded by

$$\Pr\left[E(G', \boldsymbol{P})\right] = \frac{\mathrm{O}(n^{-c})}{\Theta\left(n^{-1/2}\right)} = \mathrm{O}\left(n^{-c+1/2}\right). \qquad \square$$

3.3.5 Neighborhoods

Section 3.2 already describes some basic properties of neighborhoods in strongly hyperbolic unit disk graphs, which also hold in hyperbolic random graphs. There, we establish that the hyperbolic distance between two vertices with fixed radii r_1 and r_2 grows with increasing angular distance between them. Recall that we use $\theta(r_1, r_2)$ to denote the maximum angular distance such that they are still adjacent. Lemma 3.4 gives bounds on $\theta(r_1, r_2)$ that hold for all values of $R > 0$. Additionally, there are tighter bounds that hold asymptotically (recall that R grows logarithmically in n) [GPP12; Kro16]. In particular, [Kro16, Lemma 3.2] states that

$$\theta(r_1, r_2) = 2e^{(R-r_1-r_2)/2}\left(1 \pm \Theta\left(e^{R-r_1-r_2}\right)\right), \tag{3.10}$$

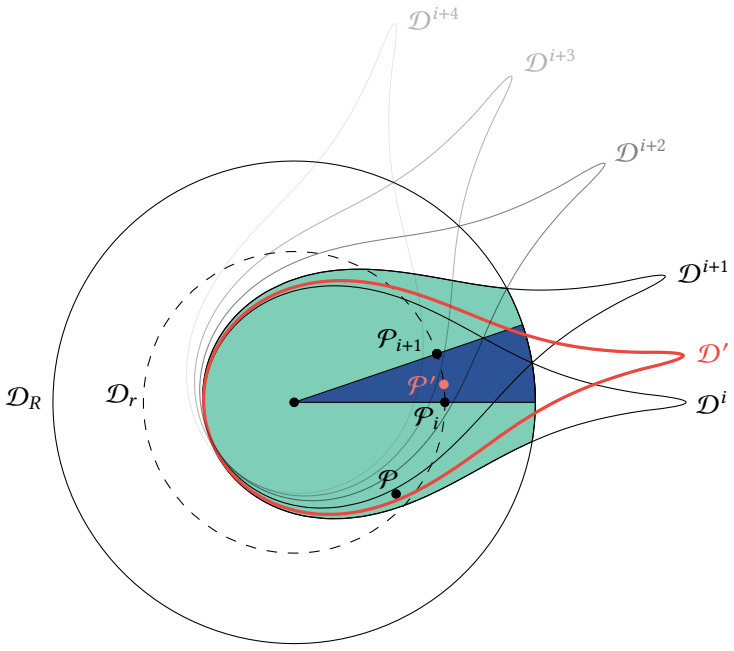


Figure 3.6: Visualization of the proof of Lemma 3.10. When constrained to the disk $\mathcal{D}_R$, the disk $\mathcal{D}'$ (red) with center $\mathcal{P}'$ at radius r is completely contained in two consecutive disks $\mathcal{D}^i$ and $\mathcal{D}^{i+1}$ (green and blue regions). Point $\mathcal{P}_i$ is between $\mathcal{P}$ and $\mathcal{P}'$.

assuming $r_1 + r_2 \geq R$. Otherwise, we have $r_1 + r_2 < R$, meaning two vertices with these radii are adjacent, independent of their angular distance.

Finally, the following lemma shows that statements about the neighborhood of a vertex with fixed angular coordinate can be extended to hold for arbitrary angular coordinates, with a small penalty in certainty. While we cannot take a union bound over infinitely many possible angular coordinates, the basic idea of the proof is to discretize the disk $\mathcal{D}_R$ into finitely many regions on which the union bound can then be applied.

Lemma 3.10. *Let G be a hyperbolic random graph. For $w \in V$, let $X_w \geq 0$ be random variables, and for $\mathcal{D} \subseteq \mathcal{D}_R$ let $X(\mathcal{D}) = \sum_{w \in \mathcal{D}} X_w$. Further, let $\boldsymbol{D}_R(r)$ be the set of disks of radius R with center at radius r. If for each $\mathcal{D} \in \boldsymbol{D}_R(r)$ it holds that $\Pr\big[X(\mathcal{D}) \leq f(n)\big] \geq 1 - p$, then*

$$\Pr\big[\forall \mathcal{D} \in \boldsymbol{D}_R(r) \colon X(\mathcal{D}) \leq 2f(n)\big] \geq 1 - \mathrm{O}(np).$$

Proof. Let $\mathcal{D}' \in \boldsymbol{D}_R(r)$ be a disk with radius R centered at radius r and arbitrary angular coordinate. To bound $X(\mathcal{D}')$, we cover the disk $\mathcal{D}_R$ with a circular sequence of n' disks $\mathcal{D}^1, \dots, \mathcal{D}^{n'}$, such that $\mathcal{D}'$ is completely contained in two consecutive disks (when constrained to the whole disk $\mathcal{D}_R$). That is, there exists an $i \in \{1, \dots, n'\}$ such that $\mathcal{D}' \cap \mathcal{D}_R \subseteq \mathcal{D}^i \cup \mathcal{D}^{i+1}$. Since $X_w \geq 0$ for all $w \in V$, it then holds that

$$X(\mathcal{D}') = \sum_{w \in \mathcal{D}'} X_w \leq \sum_{w \in \mathcal{D}^i \cup \mathcal{D}^{i+1}} X_w \leq \sum_{w \in \mathcal{D}^i} X_w + \sum_{w \in \mathcal{D}^{i+1}} X_w = X(\mathcal{D}^i) + X(\mathcal{D}^{i+1}).$$

Since $\Pr\big[X(\mathcal{D}) \leq f(n)\big] \geq 1 - p$ holds for each $\mathcal{D} \in \boldsymbol{D}_R(r)$, we can apply the union bound to conclude that $X(\mathcal{D}^i) \leq f(n)$ holds for all $i \in \{0, \dots, n'\}$ with probability $1 - n'p$. Consequently, $X(\mathcal{D}') \leq 2f(n)$ with probability $1 - n'p$.

To complete the proof, it remains to show that there exists such a sequence $\mathcal{D}^1, \dots, \mathcal{D}^{n'}$ with $n' \in \mathrm{O}(n)$. See Figure 3.6 for an illustration of how the sequence is constructed. All disks $\mathcal{D}^i$ for $i \in \{1, \dots, n'\}$ have their center at radius r. The center of the first disk is placed at angular coordinate 0 and each subsequent disk is placed at an angular distance of $2\theta(r, R)$ (see Equation (3.4)) to its predecessor until the whole disk is covered. Note that, as a consequence, the boundaries of two consecutive disks intersect at the boundary of the whole disk $\mathcal{D}_R$.

Let $\mathcal{P}'$ be the center of $\mathcal{D}'$. To see that $\mathcal{D}'$ is contained in two consecutive disks $\mathcal{D}^i$ and $\mathcal{D}^{i+1}$ (when constrained to the whole disk $\mathcal{D}_R$), first note that there exists an $i \in \{1, \dots, n'\}$ such that $\mathcal{P}'$ is between the centers $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$ of two consecutive disks $\mathcal{D}^i$ and $\mathcal{D}^{i+1}$. We show that any point $\mathcal{P} \in \mathcal{D}'$ is contained in $\mathcal{D}^i \cup \mathcal{D}^{i+1}$. Clearly, $\mathcal{D}^i \cup \mathcal{D}^{i+1}$ contains all points between $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$ (blue region in Figure 3.6). For the case where $\mathcal{P}$ does not lie between $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$, assume without loss of generality, that $\mathcal{P}_i$ is between $\mathcal{P}$ and $\mathcal{P}'$, as depicted in Figure 3.6. Since $\delta_{\mathbb{H}^2}(\mathcal{P}, \mathcal{P}') \leq R$ and since $\mathcal{P}'$ and $\mathcal{P}_i$ have the same radius but $\mathcal{P}_i$ is between $\mathcal{P}$ and $\mathcal{P}'$, it follows that $\delta_{\mathbb{H}^2}(\mathcal{P}, \mathcal{P}_i) \leq R$, and thus $\mathcal{P} \in \mathcal{D}^i$.

Finally, it remains to show that $n' = \mathrm{O}(n)$ disks are sufficient to cover the whole disk $\mathcal{D}_R$. Since two consecutive disks are placed at an angular distance of $2\theta(r, R)$, we need $n' = 2\pi/(2\theta(r, R)) = \mathrm{O}\big(1/\theta(r, R)\big)$ disks. Further, since $\theta(r, R) \geq \theta(R, R)$, it follows that $n' = \mathrm{O}\big(1/\theta(R, R)\big) = \mathrm{O}\big(e^{R/2}\big)$ due to Equation (3.10). Substituting $R = 2\log(n) + C$ then yields the claim. □

4 Routing in Strongly Hyperbolic Unit Disk Graphs

This chapter is based on joint work with Thomas Bläsius, Tobias Friedrich, and Daniel Stephan [Blä+23]. The foundation was laid as part of the master thesis of Daniel Stephan, where it was shown that there exists a routing scheme that allows for routing on hyperbolic random graphs with stretch 5, while storing $O(\log(n)^4)$ bits per vertex, asymptotically almost surely. Here, we extend this work by generalizing the results to strongly hyperbolic unit disk graphs and introducing parameters that can be used to adjust the trade-off between stretch and coordinate size, leading to an improved stretch bound, while simplifying the corresponding proofs.

4.1 Introduction

The internet is arguably the most important communication network in the world and while it is used for a variety of tasks like messaging, video streaming, gaming, and sharing resources, they all rely on a fundamental process: routing information from one network participant to another. This is done using a *routing scheme*, which is given two vertices in a graph and then tries to find a path between them.

While path finding is one of the simplest graph problems, the internet is a decentralized network, which does not allow for the use of a central data structure. Instead, each vertex can only use local information to perform a *routing decision*, i.e., the decision to which vertex the information is forwarded next such that it eventually reaches the target. This situation is further complicated by the fact that the internet consists of billions of vertices. In order to be able to handle a network of this scale, a routing scheme has to be optimized with respect to three criteria. The first two are the *space requirement* (the amount of information that the scheme uses to forward information) and the *query time* (the time it takes to make a routing decision). The third criterion is the *stretch*, i.e., how much longer the routed path is compared to a shortest path in the network. More formally, if a path between two vertices is at most c times longer than a shortest path between

them, it has *multiplicative stretch* $c \geq 1$. If it has at most at most d more vertices than a shortest path, it has an *additive stretch* of d. Note that obtaining small stretch bounds becomes harder the closer two vertices are in the graph, even if the routed path is not much longer than a shortest path in absolute terms. A better representation of the performance of the routing scheme is thus obtained by considering short detours separately. A *multiplicative stretch c with additive bound d* denotes that the routed paths have multiplicative stretch c *or* additive stretch d. Note that this implies a multiplicative stretch of $\max\{c, 1 + d\}$.

There is a long line of research on how to obtain efficient routing schemes with respect to these criteria, together with tight bounds on the trade-offs that have to be made to optimize one criterion over another. In particular, it was shown that routing with a stretch of $c \geq 1$ (i.e., allowing that paths are c times longer than the shortest paths) requires a total storage of $\Omega\left(n^{1+1/(2c+4)}\right)$ bits in general graphs [PU89].

In this chapter, we show that this lower bound can be beaten by making use of the hierarchical structures observed in graphs like the internet. To this end, we utilize strongly hyperbolic unit disk graphs, a graph class that intrinsically captures hierarchical network structures, as introduced in Section 3.2. In particular, we first formalize a framework of routing schemes by combining existing techniques like greedy routing with respect to tree-covers [FPW09; Hou+14], and distance labeling schemes [Fre+17], with an adaptation of an approach that iteratively reduces a graph [Coh98] while descending in the hierarchy. Our analysis on strongly hyperbolic unit disk graphs shows that stretch, space requirement, and query time can be adjusted using the parameters of the scheme. In particular, for hyperbolic random graphs, which have been shown to represent graphs like the internet well [BPK10], we obtain a routing scheme with stretch 3, while asymptotically almost surely using $\mathrm{O}\left(n \log(n)^4\right)$ bits of storage in total and answering queries in $\mathrm{O}\left(\log(n)^2\right)$ time. Finally, a brief empirical evaluation indicates that our results translate well to real-world networks.

4.2 A Brief History of Routing Schemes

In the following, we summarize the main approaches to adjusting the trade-off between stretch and required space. The query times of the considered schemes are at most polylogarithmic. Figure 4.1 gives an overview of existing schemes.

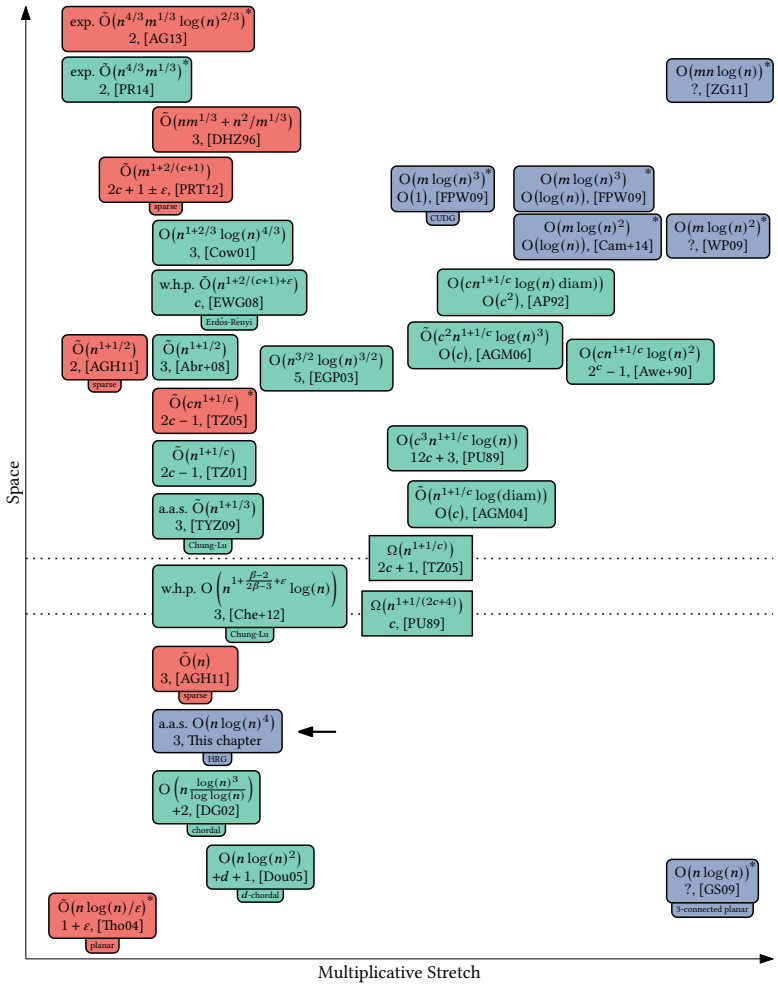


Figure 4.1: Distance oracles (red), routing schemes (green), and local routing schemes (blue) arranged by space requirements (first line) and multiplicative stretch (second line). Additive stretch is denoted with a preceding +. Lower bounds are shown with rectangular corners. We note that in order to be able to compare different results, we unified the representation of the required space. That is, when the bounds on the space only account for the size of the considered data structures but not its content or only bound the space required for a single vertex, we adjusted the bounds such that they represent the space required for all content stored for all vertices and marked the corresponding entry with an asterisk ($*$). In particular, greedy routing schemes typically require that every vertex stores the information of all its neighbors. The total space requirement is then obtained by summing over all vertex degrees, which introduces a factor of m (the number of edges in the graph) over the storage required to address a single vertex.

4.2.1 Routing Schemes

In general networks, routing with a stretch of 1, i.e., always routing along the shortest paths, requires storing $\Theta(n^2 \log n)$ bits in total [GP96]. The most commonly used approach to reducing the required space is to only store shortest path information for certain vertex pairs. That is, a representation of a subgraph (typically a tree or a collection of trees) of the original graph is stored and the routing takes place on the subgraph. This is usually done by selecting a set of *landmark* (or *pivot*) vertices. Then, for each vertex only the information about how to get to the closest landmark is stored [Abr+08; AGM04; AGM06; AP92; Awe+90; Cow01; EGP03; PR14; TZ01]. These schemes basically partition the graph based on the landmark vertices. A related approach starts with a partition and defines the landmarks afterwards [PU89; RT16]. The scheme then routes via the landmarks closest to the source and target. This general approach can be optimized in several ways. First, the network can be partitioned with several levels of granularity, such that messages that need to travel larger distances are routed to landmarks whose associated vertex set is larger [Awe+90; PU89]. Improvements for shorter distances can be obtained by storing the actual shortest path information for vertices in close vicinity of each vertex [Awe+90; EGP03]. Moreover, the selection of the landmarks itself can have an impact on the routing performance. In general graphs they are typically selected at random. A more careful selection can lead to better results on Erdős-Rényi random graphs [EWG08], or when assuming that the network has certain properties like a power-law degree distribution [Che+12; TYZ09]. Similarly, better results can be obtained on chordal graphs [DG02; Dou05; DYL04].

Closely related to routing schemes are *approximate distance oracles*. Virtually the only difference is that, there, we are only interested in the length of a short path instead of the path itself. As before, the most commonly used techniques are based on landmarks [AG13; DHZ96; PRT12; TZ05], and compared to general networks, better results can be obtained when assuming that the considered graphs have certain properties like being planar [Tho04], or being sparse (although at the expense of an increased query time) [AGH11].

In general, routing schemes and distance oracles are based on one central data structure that holds the information required for routing. This can become an issue with increasing network size, since achieving a stretch of c requires a data structure of size $\Omega\left(n^{1+1/(2c+4)}\right)$ on general graphs [PU89]. One approach to overcoming this problem is to consider local routing schemes instead.

4.2.2 Local and Greedy Routing Schemes

In local routing schemes the routing information is distributed and each vertex can only use its own information to forward messages. One approach to achieving this are *interval routing schemes*, where each vertex is equipped with a mapping from its outgoing edges to a partition of the vertices in the graph and the message is forwarded along the edge whose assigned vertex set contains the target [EGP03; PU89; SK85].

Another popular approach is *geographic* or *greedy* routing. There, each vertex is assigned a coordinate in a metric space and a message is routed to a neighbor that is closer to the target with respect to the metric. While initially being motivated by real-world networks with actual geographic locations [KK00; TK84], later adaptations assigned *virtual coordinates* [Rao+03].

In addition to the previously mentioned criteria, greedy routing is also evaluated regarding the *success rate*, since the virtual coordinates may be assigned such that forwarding messages greedily leads to a dead end. Even simple graphs like a star with six leaves cannot be embedded in the Euclidean plane such that greedy routing always succeeds [PR05]. Worse yet, even if a graph admits a greedy embedding into the Euclidean plane, there are graphs that require $\Omega(n)$ bits per coordinate [ADF12]. However, it was shown that delivery can be guaranteed on every graph when embedding it in the hyperbolic plane [Kle07]. Unfortunately, due to the properties of hyperbolic space, this requires high-precision coordinates, which leads to an increased space requirement [Blä+20]. While attempts have been made to reduce the coordinate size [EG08], it has been shown that this remains an open problem [Blä+20; KK15]. However, in many greedy routing schemes the space per coordinate is at most polylogarithmic [BC06; Cam+14; EG11; GS09; Muh07; WP09; ZG11].

Unfortunately, not much is known about stretch in local routing schemes. There are bounds in the order of $O(\log n)$ for general graphs [Cam+14; FPW09], and on graphs of bounded hyperbolicity, we can use a distance labeling scheme (the local equivalent to approximate distance oracles) to obtain a greedy routing scheme with an additive stretch of $O(\log n)$ [GL05]. However, it has been observed that greedy routing schemes can achieve much better stretch in practice, which we discuss in the following.

4.2.3 Routing in Practice

Real-world networks rarely resemble the worst cases considered in the previously mentioned results. More realistic insights can be obtained by analyzing networks whose properties resemble those of real-world graphs, like the small-world phenomenon [Kle00]. The previously mentioned Euclidean unit disk graphs (Chapter 1) are used to model wireless sensor networks. In such graphs greedy routing can be performed with constant stretch [FPW09], compared to logarithmic in general graphs. Additionally, better trade-offs between stretch and space have been obtained on sparse graphs [AGH11; PRT12], and Chung-Lu random graphs [ACL01]. There, the best known space bound of $\tilde{\mathrm{O}}\big(n^{1+1/2}\big)$ for a stretch of 3 on general graphs [TZ01], was improved to $\mathrm{O}\big(n^{1+(\beta-2)/(2\beta-3)+\varepsilon}\big)$, with high probability, for power-law exponent $\beta \in (2,3)$ and $\varepsilon > 0$ [Che+12]. Experiments on internet-like networks further indicate that the landmark-based routing schemes due to Thorup and Zwick [TZ01] yield a rather low stretch of about 1.1 while the information stored at the vertices is small as well [KFY04; Kri+07]. Similar results have been obtained in experiments on internet topologies and random graphs with power-law degree distributions [AGH11; Che+12; TYZ09].

Additionally, it was observed that greedy routing works remarkably well on internet graphs, when assuming an underlying hyperbolic geometry. There, a network is embedded into the hyperbolic plane and a message is always forwarded to the neighbor with the smallest hyperbolic distance to the target. While delivery is not guaranteed in the resulting scheme, experiments show that it achieves success rates of at least 97% and a stretch of about 1.1 on internet topologies [BPK10; Pap+10]. Partly motivated by these results Krioukov et al. introduced the hyperbolic random graph (HRG) model that is used to represent real-world networks like the internet (see Chapter 3). For a generalized version of this model it was already shown that a greedy routing procedure that succeeds with constant probability can almost surely achieve an average stretch of $1 + o(1)$ [Bri+17]. Nevertheless, it was unclear whether, in a addition to the small stretch, greedy routing on realistic representations of internet-like graphs can be implemented, such that delivery is always guaranteed, while keeping the space requirement low. In this chapter, we answer this question by developing a greedy routing scheme that always succeeds with small stretch. Additionally, the space requirement is small on networks with underlying hyperbolic geometry.

4.3 Greedy Routing

The basic idea of greedy routing is to always forward a message to a neighbor of a vertex that is closer to the target. We note that the term *greedy* is often used to refer to a scheme that routes to a neighbor that is *closest* to the target. However, for our purposes this constraint is not required. When designing a greedy routing scheme, we need to compute distances between vertices and select a suitable neighbor with respect to these distances. Moreover, both processes need to be done quickly and with as little information as possible, in order to keep the query time and the space requirement low.

In the following, we first define a distance function by combining graph distances of subgraphs with bounded stretch and show that routing with respect to this function yields the same stretch. Afterwards, we explain how the function can be computed efficiently using distance labeling schemes. A simple greedy routing scheme is then obtained by storing at each vertex the labels of its neighbors and, in a routing step, computing the distances from all of them to the target in order to find one that is closer than the current vertex. We then reduce the required space and query time by extending the labels in a way that allows us to quickly identify a suitable neighbor without considering all of them. Finally, these insights are combined with the fact that there exist efficient labeling schemes for trees, to obtain an efficient greedy routing scheme.

4.3.1 Combining Graph Distances

Let $G = (V, E)$ be a graph and let $\delta\colon V \times V \to \mathbb{R}_{\geq 0}$ be a semi-metric on G. That is, for all $s, t \in V$ we have $\delta(s, t) \geq 0$, $\delta(s, t) = 0$ if and only if $s = t$, and $\delta(s, t) = \delta(t, s)$. A greedy routing scheme routes *with respect to* δ, if at s a message to t is forwarded to a neighbor v of s where $\delta(v, t) < \delta(s, t)$. Note that, depending on δ such a neighbor may not exist and the message cannot be forwarded, which is called *starvation*. In contrast, a routing scheme with guaranteed delivery is called *starvation-free*. It is known that greedy routing is starvation-free, if at every vertex $s \neq t$ there is a neighbor v with $\delta(v, t) < \delta(s, t)$ (see, e.g., [ZG13]).

We say that δ is *integral* if it maps to the natural numbers, i.e., $\delta\colon V \times V \to \mathbb{N}$. Note that, if δ is integral and routing with respect to δ is starvation-free, the distance to the target decreases by at least one in each step. Thus, the length of

the routed path between s and t is bounded by $\delta(s,t)$. When this is the case, we say that routing with respect to δ is *δ-bounded*.

Given a connected graph G, a natural choice for determining a distance between s and t is to use the length of a shortest path between them, which we denote by $\delta_G(s,t)$. Note that δ_G is integral and that the successor of s on a shortest s-t-path is a neighbor of s and is closer to t. Thus, routing with respect to δ_G yields perfect stretch. However, δ_G cannot be computed while simultaneously keeping the required space and query time low (see Section 4.2). Therefore, we relax the constraint on routing with respect to exact graph distances and use upper bounds instead. This can be achieved by taking a subgraph G' of G and routing on G with respect to $\delta_{G'}$. The stretch of the resulting routing scheme depends on how well the distances in G' approximate the distances in G. Unfortunately, finding a subgraph with good stretch is hard in general [Cai94; PS89]. However, instead of routing with respect to the distances in a single subgraph, we can combine the distances in multiple subgraphs. To obtain a good stretch, it then suffices to find low-stretch subgraphs for small parts of the graph.

A collection of graphs $\boldsymbol{C}$ is a (c,d,k)-*graph-cover* of $G=(V,E)$, if for all $s,t\in V$ there exists a connected subgraph G' of G in $\boldsymbol{C}$ with $\delta_{G'}(s,t)\leq c\cdot\delta_G(s,t)$ or $\delta_{G'}(s,t)\leq\delta_G(s,t)+d$, and every vertex $v\in V$ is contained in at most k graphs in $\boldsymbol{C}$. We say that $\boldsymbol{C}$ has multiplicative stretch c with additive bound d. For two vertices s and t we define $\delta_{\boldsymbol{C}}(s,t)=\min_{G'\in\boldsymbol{C}}\delta_{G'}(s,t)$.

Lemma 4.1. *Let G be a graph and let $\boldsymbol{C}$ be a (c,d,k)-graph-cover of G. Then, greedy routing on G with respect to $\delta_{\boldsymbol{C}}$ has multiplicative stretch c with additive bound d.*

Proof. Let $s\neq t$ be two vertices. To prove the claim, we need to show that an s-t-path obtained by greedily routing with respect to $\delta_{\boldsymbol{C}}$ has length at most $c\cdot\delta_G(s,t)$ or $\delta_G(s,t)+d$. To this end, we prove that the resulting routing scheme is $\delta_{\boldsymbol{C}}$-bounded. The claim then follows, due to the fact that consequently the routed s-t-path has length at most $\delta_{\boldsymbol{C}}(s,t)=\min_{G'\in G}\delta_{G'}(s,t)$ and the fact that there exists a $G'\in G$ with $\delta_{G'}(s,t)\leq c\cdot\delta_G(s,t)$ or $\delta_{G'}(s,t)\leq\delta_G(s,t)+d$ by assumption.

Since $\delta_{\boldsymbol{C}}$ is the minimum of integral semi-metrics, it is itself an integral semi-metric. Therefore, it suffices to show that routing with respect to $\delta_{\boldsymbol{C}}$ is starvation-free, which is the case, if for every two vertices $s\neq t$ there exists a neighbor v of s in G with $\delta_{\boldsymbol{C}}(v,t)<\delta_{\boldsymbol{C}}(s,t)$. Consider the connected subgraph

$G' \in \boldsymbol{C}$ for which $\delta_{G'}(s,t) = \delta_C(s,t)$. Then, there exists a shortest path from s to t in G'. For the successor v of s on this path, it holds that $\delta_{G'}(v,t) = \delta_{G'}(s,t) - 1$ and thus $\delta_C(v,t) \leq \delta_{G'}(s,t) - 1 = \delta_C(s,t) - 1 < \delta_C(s,t)$. Finally, since G' is a subgraph of G, it follows that v is also a neighbor of s in G. □

In order to show that δ_C can be computed efficiently, we use *distance labeling schemes* [Gav+04]. A distance labeling scheme implements a semi-metric δ by assigning each vertex a *distance label*, such that for two vertices s, t we can compute $\delta(s,t)$ by looking at their distance labels only. The *label size* of a distance labeling scheme denotes the maximum number of bits required to represent the label of a vertex. The *query time* is the time it takes to compute δ using the labels. Given a graph-cover $\boldsymbol{C}$, we obtain a distance labeling scheme that implements δ_C by combining distance labeling schemes for the contained subgraphs.

Lemma 4.2. *Let G be a graph and let $\boldsymbol{C}$ be a (c,d,k)-graph-cover of G such that for every $G' \in \boldsymbol{C}$ there exists a distance labeling scheme that implements $\delta_{G'}$ with label size ℓ and query time q. Then, there exists a distance labeling scheme for G that implements δ_C with label size $\mathrm{O}\big(k(\ell + \log k + \log n)\big)$ and query time $\mathrm{O}\big(kq\big)$.*

Proof. We assign each subgraph $G' \in \boldsymbol{C}$ a unique *graph-ID* in $[|\boldsymbol{C}|]$ and compute the distance labels for all vertices in G'. By combining the distance labels with the corresponding graph-ID, we obtain an *identifiable distance label* that can be used to uniquely identify to which graph a distance label belongs. The label of a vertex v is then obtained by collecting the identifiable distance labels of v for all subgraphs that v is contained in and sorting them by graph-ID.

The label size can now be bounded as follows. Since each vertex v is contained in at most k subgraphs, we can conclude that $|\boldsymbol{C}| \leq kn$. Therefore, the graph-IDs can be encoded using $\mathrm{O}\big(\log k + \log n\big)$ bits. Moreover, by assumption the distance labels in the subgraphs can be represented using ℓ bits. It follows that a single identifiable distance label takes $\mathrm{O}\big(\ell + \log k + \log n\big)$ bits. Again, since every vertex is contained in at most k subgraphs, v's label consists of at most k identifiable distance labels. Consequently, the label size is bounded by $\mathrm{O}\big(k(\ell + \log k + \log n)\big)$.

It remains to bound the query time. Given the collection of identifiable distance labels of two vertices, we can identify the ones with matching graph-IDs in time $\mathrm{O}(k)$, since they are sorted by graph-ID. For each match we compute the distance in the corresponding subgraph in time q. Afterwards the minimum distance can be found in $\mathrm{O}(k)$ time. It follows that δ_C can be computed in time $\mathrm{O}\big(kq\big)$. □

4.3.2 Finding a Suitable Neighbor

Given a distance labeling scheme, a simple greedy routing scheme is obtained by storing at each vertex s the distance label of s, as well as the ones of all neighbors of s. Using the distance label of a target t, we can compute the distances between t and all neighbors of s, in order to find one that is closer to t. Compared to the labeling scheme, the required space and the query time at s then increase by a factor of $\deg(s)$. This dependence on the degree can be undesirable as many real-world networks contain vertices with large degrees for which the label size would be large as well. This can be avoided by utilizing another kind of label that allows us to compute the successor of s on an s-t-path instead of the length of this path.

First note that we need to be able to identify a neighbor directly, if we want to avoid performing a linear search over all neighbors. To this end, we assign each neighbor v of s a unique *port* $p_s(v) \colon N(s) \to \{1, \dots, n\}$. Finding a neighbor of s that is closer to a target t with respect to a semi-metric δ then boils down to determining the corresponding port. To this end, we can use a *port labeling scheme* that implements δ. Such a scheme assigns each vertex in a graph a *port label* such that we can determine the port of a neighbor of s that is closer to t with respect to δ, by only looking at the port labels of s and t. The corresponding label sizes and query times are defined analogous to how they are defined for distance labels.

In the following we show how, given a graph-cover $\boldsymbol{C}$, we can combine distance and port labels of the subgraphs in the cover, to obtain a port labeling scheme that implements δ_C.

Lemma 4.3. *Let G be a graph and let $\boldsymbol{C}$ be a (c, d, k)-graph-cover of G such that for every $G' \in \boldsymbol{C}$ there exist distance and port labeling schemes that implement $\delta_{G'}$ with label size ℓ and query time q. Then, there exists a port labeling scheme for G that implements δ_C with label size $\mathrm{O}\big(k(\ell + \log k + \log n)\big)$ and query time $\mathrm{O}\big(kq\big)$.*

Proof. For every vertex s in G we fix a port assignment for the neighbors of s. Afterwards, we assign the same ports in the subgraphs G' of G in $\boldsymbol{C}$ that s is contained in. More precisely, if v is a neighbor of s in G', then the port $p_s(v)$ is identical in G and G'. As a consequence, we can use a port labeling scheme in G' to determine the port of a neighbor of s in G.

Now consider the distance labeling scheme described in Lemma 4.2, where we assign each subgraph $G' \in \boldsymbol{C}$ a unique graph-ID and compute distance labels

for all vertices in all subgraphs to obtain identifiable distance labels. In addition, we now compute port labels for all vertices in all subgraphs. By combining them with the previously obtained identifiable distance labels, we obtain *identifiable distance port labels*. As before, the label of a vertex v then consists of the collection of identifiable distance port labels of v in all subgraphs that v is contained in, sorted by graph-ID.

We continue the proof by showing that, given the labels of two vertices $s \neq t$, we can compute the port of a neighbor of s in G that is closer to t with respect to δ_C. As described in the proof of Lemma 4.2, we can use the labels to find the graph-ID of a subgraph G' of G for which $\delta_{G'}(s,t) = \delta_C(s,t)$. We then use the corresponding port labels of s and t to determine the port $p_s(v)$ of a neighbor v of s that is closer to t with respect to $\delta_{G'}$. Clearly, we have $\delta_C(v,t) \leq \delta_{G'}(v,t) < \delta_{G'}(s,t) = \delta_C(s,t)$. Moreover, since $p_s(v)$ is identical in G' and G, it follows that $p_s(v)$ is a suitable port in G.

The label size can be bounded as follows. By Lemma 4.2 we can encode all k identifiable distance labels stored at a vertex using $O\big(k(\ell + \log k + \log n)\big)$ bits. Since a single identifiable distance label is extended with a port label that takes at most ℓ bits, it follows that the label size increases by an additive $O(k\ell)$, yielding a size of $O\big(k(\ell + \log k + \log n)\big)$ bits.

To finish the proof, it remains to bound the query time. Again, as described in the proof of Lemma 4.2, determining the graph-ID of the subgraph G' for which $\delta_{G'}(s,t) = \delta_C(s,t)$ takes $O\big(kq\big)$ time. Computing the port $p_s(v)$ of a suitable neighbor v of s then takes an additional time q. Consequently, we obtain a query time of $O\big(kq\big)$ in total. □

4.3.3 Tree-Cover-Based Greedy Routing

We are now ready to combine the above results to obtain our greedy routing scheme. To this end, we need to find (c,d,k)-graph-covers with small values for c, d, and k, as well as distance and port labeling schemes with small label sizes and query times, as all of these properties affect the performance of the routing scheme. While distance labeling schemes require large labels in general graphs [Gav+04], better results can be obtained by restricting the graph-cover to only contain trees as subgraphs. Such a cover is then called *tree-cover*. Tree-covers are standard in routing [AKP94; AP92; EGP03; FPW09; Hou+14; Tan+10; TZ01], and while it is known that greedy routing with respect to δ_C for a (c,d,k)-tree-cover $\boldsymbol{C}$ is starvation-free (see e.g., [Hou+14]), we also know that the

resulting routing scheme has stretch c with additive bound d due to Lemma 4.1. Moreover, for trees there are distance and port labeling schemes with $O(\log(n)^2)$ bit labels and constant query time [Fre+17; TZ01]. Together with Lemma 4.3 we obtain the following theorem.

Theorem 4.4. *Let G be a graph. Given a (c, d, k)-tree-cover of G, greedy routing on G can be implemented such that the resulting routing scheme is starvation-free, has stretch c with additive bound d, stores $O(k(\log(n)^2 + \log k))$ bits at each vertex, and takes $O(k)$ time for a routing decision.*

4.4 Tree-Cover Algorithm

With the basic framework set up, it remains to find tree-covers with bounded stretch. To this end, we propose an algorithm that is an adaptation of a previously proposed algorithm for computing graph spanners [Coh98]. The idea is to compute a tree-cover of a graph using repeated breadth-first searches (BFS) while removing parts of the graph after each search. On the one hand, utilizing BFS-trees allows us to ensure that the distances between vertices in the trees of the cover are not much larger than the distances in the original graph. On the other hand, removing vertices while processing the graph ensures that the number of trees a vertex is contained in is small. More precisely, we introduce two parameters a and b that can be used to tune both aspects. Throughout the procedure the number of levels explored in a single BFS grows exponentially, where b is the base of the exponent and a is part of an additional factor. The larger b, the quicker grows the portion of a tree that gets removed after a BFS, which also increases the graph distances in the remaining graph more quickly than when using a small value for b. As a consequence, the distances captured in succeeding BFSs are more likely to deviate from shortest paths in the original graph, which increases the stretch. To counteract this, we can choose a large, which increases the depth of the BFS tree without affecting the part that gets removed. Then vertices appear together in more trees, making it more likely that a short path between them is captured by one of the trees. The resulting stretch, thus, depends on the choice of a and b, but is independent of the considered input graph.

Before presenting the algorithm, we first characterize a setting in which we can easily bound the stretch of a subtree of a graph.

Lemma 4.5. *Let $G = (V, E)$ be a graph, $u, v \in V$, and let H be an induced subgraph that contains all vertices on a shortest uv-path P in G. Let T be a partial shortest-path tree in H rooted at t that contains u and v. Then, for every vertex w in T that lies on P, $\delta_T(u,v) \leq \delta_G(u,v) + 2\delta_H(t,w)$.*

Proof. Let t' be the lowest common ancestor of u and v and consider the paths P_u and P_v from t' to u and v, respectively. Note that P_u and P_v are shortest paths in H as they are descending paths in a shortest-path tree. Thus,

$$\delta_T(u,v) = |P_u| + |P_v| = \delta_H(t',u) + \delta_H(t',v).$$

Observe that clearly $\delta_H(t',u) \leq \delta_H(t,u)$. Moreover, by the triangle inequality, we have $\delta_H(t,u) \leq \delta_H(t,w) + \delta_H(w,u)$. Analogously for v, we obtain $\delta_H(t',v) \leq \delta_H(t,w) + \delta_H(w,v)$. Thus, we get

$$\begin{aligned}\delta_T(u,v) &= \delta_H(t',u) + \delta_H(t',v) \leq \delta_H(t,w) + \delta_H(w,u) + \delta_H(t,w) + \delta_H(w,v)\\ &= \delta_H(u,v) + 2\delta_H(t,w),\end{aligned}$$

where the last equality holds as w lies on a shortest uw-path P in G, which is also a shortest uw-path in H, since H is an induced subgraph of G that contains all vertices of P. For the same reason, we get $\delta_H(u,v) = \delta_G(u,v)$, which proves the claim. □

Consider the setting as in the above Lemma, let w be chosen such that $\delta_H(t,w)$ is minimal, and let $\xi = 2\delta_H(t,w)/\delta_G(u,v)$. Then, we can bound the distance between u and v in T by $\delta_T(u,v) \leq (1+\xi)\delta_G(u,v)$. That is, T has stretch $(1+\xi)$. The following algorithm computes a tree-cover with the same stretch.

Let G be the input graph. The algorithm operates in phases, starting with phase 0. For each phase i, we define a radius $r_i = b^i$, for a base $b > 1$. Then, for $a > 0$, we choose a vertex t in the current graph and compute the partial shortest-path tree with root t containing all vertices with distance at most $(1+a)r_i$ from t. Afterwards, we delete all vertices with distance at most r_i to t from the current graph. This is iterated until all vertices are deleted. Afterwards, phase i is done and we restore the original input graph G before starting phase $i+1$. This process is stopped, once the whole graph is deleted after processing the first tree in a phase. The output of the algorithm is the set of all computed trees. Since the algorithm **pro**duces **t**ree-covers **o**f **n**etworks, we call it Proton.

Note that Proton has several degrees of freedom. We can choose the parameters $a > 0$ and $b > 1$, as well as the order in which the roots of the partial shortest-path trees are selected. The following lemma holds independent of the root selection strategy.

Lemma 4.6. *The tree-cover computed by Proton has stretch* $(1 + 2b/a)$ *with additive bound* 2.

Proof. Let $\boldsymbol{C}$ be the tree-cover computed by Proton, let $G = (V, E)$ be the input graph, and let $u \neq v \in V$ be two arbitrary vertices. We have to show that $\boldsymbol{C}$ contains a tree T that includes u and v such that $\delta_T(u,v) \leq (1 + 2b/a)\delta_G(u,v)$ or $\delta_T(u,v) \leq \delta_G(u,v) + 2$.

Let i be minimal such that $\delta_G(u,v) \leq ar_i$. Assume for now that Proton did not stop before phase i; we deal with the other case later. As phase i continues until all vertices are deleted, at one point a vertex w on a shortest uv-path in G is deleted. Let H be the current graph before that happens for the first time and let T be the partial shortest-path tree computed in H rooted at t. To show that T is the desired tree, we aim to apply Lemma 4.5.

First note that H is an induced subgraph of G that contains all vertices on a shortest uw-path of G. Moreover, T contains u and v for the following reason. As w is deleted, we know that $\delta_H(t,w) \leq r_i$. Moreover, as w lies on a shortest uv-path, the distance from w to either u or v cannot exceed $\delta_H(u,v) = \delta_G(u,v)$. Thus, by the triangle inequality and the above choice of i, we have $\delta_H(t,u) \leq \delta_H(t,w) + \delta_H(w,u) \leq r_i + ar_i = (1 + a)r_i$, which implies that u is a vertex of T. Analogously, v is also contained in T.

With this, we can apply Lemma 4.5, yielding a stretch of $(1 + \xi)$ where ξ is given by $\xi = 2\delta_H(t,w)/\delta_G(u,v)$. To bound ξ, recall that we chose i minimal such that $\delta_G(u,v) \leq ar_i$. Thus, if $i > 0$, then $\delta_G(u,v) > ar_{i-1} = a/b \cdot r_i$. Together with the fact that $\delta_H(t,w) \leq r_i$, we obtain $\xi \leq 2b/a$, as desired. In the special case that $i = 0$ we have $r_i = 1$ and therefore $\delta_H(t,w) \leq 1$. Thus, Lemma 4.5 directly yields $\delta_T(u,v) \leq \delta_G(u,v) + 2$, which is covered by the additive bound 2.

Finally, we assumed above that Proton did not stop before phase i and it remains to consider the case where it stops in phase $j < i$. In this case, let T be the tree we get in phase j, which includes all vertices of G. Let t be the root of T. As all vertices have distance at most r_j from t, we get $\delta_T(u,v) \leq 2r_j$. Moreover, as i was chosen minimal such that $\delta_G(u,v) \leq ar_i$, we have $\delta_G(u,v) > ar_j$. Together with the previous inequality, this gives a stretch of $2/a$, which is smaller than the desired $(1 + 2b/a)$, as $b > 1$. □

4.5 Performance on Strongly Hyperbolic Unit Disk Graphs

As explained above, the stretch of the tree-cover obtained by Proton directly translates to our routing scheme. To further assess the performance of the scheme with respect to the storage requirements and query time, it remains to determine k, which denotes the maximum number of trees that a single vertex is contained in. In contrast to the stretch, its value *does* depend on the input graph. In the following, we consider how the algorithm performs on strongly hyperbolic unit disk graphs (see Section 3.2). As mentioned earlier, these capture hierarchical structures as often observed in real-world graphs. We can utilize this property by descending in the hierarchy as we remove parts of the graph. In the underlying hyperbolic geometry, this is reflected by moving towards the boundary of the hyperbolic disk that the vertices are contained in. That is, the roots of the BFS trees are chosen in order of increasing distance to the center of the disk in the corresponding unit disk representation of the graph. We refer to this as the *radially increasing* root selection strategy. By utilizing the fact that graph distances roughly represent hyperbolic distances between vertices, we can identify areas in the disk that cannot contribute trees to the cover without removing each other, which in turn allows us to bound k. We obtain the following theorem.

Theorem 4.7. *Let G be a strongly hyperbolic unit disk graph with radius $R > 0$. Given the disk representation of G, $a > 0$, and $b > 1$, the Proton algorithm with the radially increasing root selection strategy computes a (c, d, k)-tree-cover of G with $c = 1 + 2^{b}/_{a}$, $d = 2$, and*

$$k = \pi e\left(\frac{1+a}{b-1}(b^2 \operatorname{diam}(G) - 1)R + 2\big(\log_b(\operatorname{diam}(G)) + 2\big)\right).$$

First, note that the correctness of the claimed stretch immediately follows from Lemma 4.6. However, bounding k is more involved. In the following, we first compute an upper bound on k that holds for a given phase and afterwards sum over all phases. To this end, we analyze several aspects about paths in strongly hyperbolic unit disk graphs, depending on the positions of the vertices.

Consider the roots of the partial shortest-path trees that contain a vertex v in a given phase, which we refer to as the *roots of v*. We partition the hyperbolic disk

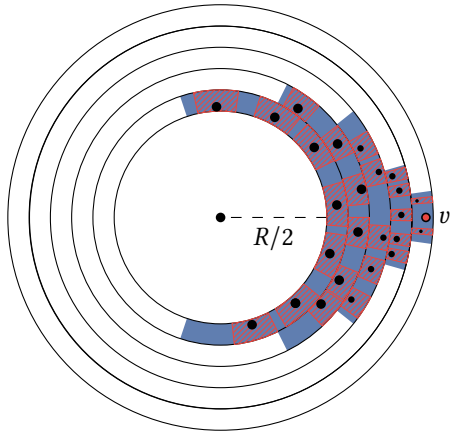


Figure 4.2: Illustration of the proof of Theorem 4.7. The hyperbolic disk is divided into bands. The roots (black vertices) of v (red vertex) in a band lie in an angular interval of bounded width (blue). Each root reserves a portion of that interval (red) that no other root can lie in. All vertices with radius at most $R/2$ are removed after processing the first root.

into annuli, which we call *bands*, and compute an upper bound on the number of roots of v in each band, see Figure 4.2 for an illustration. We then utilize two key ingredients. First, since v is contained in the partial shortest-path trees of its roots, the length of the path between v and a root is bounded, and so is the angular distance between them. Consequently, all roots in a band lie in a bounded angular interval (blue areas in Figure 4.2). Secondly, roots cannot be adjacent as they would otherwise delete each other, which means that the hyperbolic distance between them has to be sufficiently large. For roots in the same band, this can only be achieved if their angular distance is large. Consequently, each root in a band reserves a portion of the angular interval (red areas in Figure 4.2) that no other root can lie in, from which we can derive an upper bound on the number of roots that lie in the band.

The following lemma bounds the angular distance $\delta_\varphi(u, u_k)$ between a vertex u and another vertex u_k, assuming that there exists a path of length k between them that consists only of vertices whose radii are not smaller than the one of u. In particular, this applies to roots of v: In a given phase, the length of the paths considered in the partial shortest-path trees is bounded. Moreover, when the partial shortest-path tree of a root ρ of v is computed, all vertices of smaller

radii than ρ have been deleted (since roots are considered in order of increasing radius), meaning the path from ρ to v cannot contain vertices of smaller radius.

Lemma 4.8. *Let G be a strongly hyperbolic unit disk graph with radius $R > 0$ and let u be a vertex with $r(u) \geq R/2$. Further, let $P = (u, u_1, \dots, u_k)$ be a path with $r(u) \leq r(u_i)$ for all $i \in [k]$. Then, $\delta_\varphi(u, u_k) \leq k \cdot \pi e^{R/2 - r(u)}$.*

Proof. For convenience, we define $u_0 = u$. Then, $\delta_\varphi(u, u_k)$ can be bounded by

$$\delta_\varphi(u, u_k) \leq \sum_{i=1}^{k} \delta_\varphi(u_{i-1}, u_i).$$

Note that u_{i-1} and u_i are adjacent and recall that $\theta(r(u_{i-1}), r(u_i))$ denotes the maximum angular distance between them, such that this is the case (see Section 3.2.2). Thus,

$$\delta_\varphi(u, u_k) \leq \sum_{i=1}^{k} \theta(r(u_{i-1}), r(u_i)).$$

Since $R/2 \leq r(u) \leq r(u_i)$ for all $i \in [k]$ is a precondition of this lemma, we have $r(u_{i-1}) + r(u_i) \geq R$ for all $i \in [k]$. Consequently, we can apply Lemma 3.4 to bound $\theta(r(u_{i-1}), r(u_i))$, which yields

$$\begin{aligned}
\delta_\varphi(u, u_k) &\leq \sum_{i=1}^{k} \pi e^{(R - r(u_{i-1}) - r(u_i))/2} \\
&\leq \sum_{i=1}^{k} \pi e^{(R - r(u) - r(u))/2} \\
&= k \cdot \pi e^{R/2 - r(u)},
\end{aligned}$$

where the second inequality is valid since $r(u) \leq r(u_i)$ for all $i \in [k]$. □

The second key ingredient is a lower bound on the minimum angular distance between two non-adjacent vertices in a radial band of fixed width in the hyperbolic disk. We note that in order to obtain a bound that is easy to work with, we aim to utilize Corollary 3.5. However, this requires that R is not too small. For now, we assume that this requirement is met and afterwards resolve the constraint in the analysis of the algorithm.

Lemma 4.9. *Let G be a strongly hyperbolic unit disk graph with radius $R \geq 1$ and let $r \geq R/2$ be a radius. Further, let u, v be non-adjacent vertices with $r(u), r(v) \in [r, r+\tau]$ for $\tau \in [0, R-1]$. Then, $\delta_\varphi(u,v) \geq e^{R/2-(r+\tau)}$.*

Proof. Recall that $\theta(r(u), r(v))$ denotes the maximum angular distance such that u and v are adjacent. As the two vertices are not adjacent in our case, it follows that $\delta_\varphi(u,v) > \theta(r(u), r(v))$. We now aim to apply Corollary 3.5 in order to obtain a lower bound on $\theta(r(u), r(v))$. To this end, we first validate that its preconditions are met. Since $r(u), r(v) \geq r \geq R/2$, we have $r(u) + r(v) \geq R$. Moreover, by assumption we know that $r(u), r(v) \in [r, r+\tau]$ for $\tau \in [0, R-1]$, which implies that $|r(u) - r(v)| \leq \tau \leq R-1$. Consequently, we can apply Corollary 3.5 to conclude that

$$\begin{aligned} \theta(r(u), r(v)) &\geq e^{(R-r(u)-r(v))/2} \\ &\geq e^{(R-2r-2\tau)/2} \\ &= e^{R/2-(r+\tau)}, \end{aligned}$$

where the second inequality is valid, since by assumption $r(u), r(v) \leq r + \tau$. □

We can now combine the two key ingredients to compute an upper bound on the number of the roots of v in a given phase i, which we denote by $\boldsymbol{\rho}_i(v)$.

Lemma 4.10. *Let G be a strongly hyperbolic unit disk graph with radius $R > 0$. Let the disk representation of G, $a > 0$, and $b > 1$ be given and consider phase i of the* Proton *algorithm. Then, for every vertex v it holds that*

$$|\boldsymbol{\rho}_i(v)| \leq \pi e(R(1+a)b^i + 2).$$

Proof. In the following, we aim to utilize Lemmas 4.8 and 4.9, both of which require that the considered vertices have a radius of at least $R/2$ and one additionally assumes that $R \geq 1$ Therefore, we first argue about the case where these conditions are not met. First note that after the first root in a phase is processed, all vertices with radius at most $R/2$ are removed since (if they exist in the first place) they form a clique. Additionally, when $R < 1$, the whole graph can be covered by few cliques. More precisely, by Lemma 3.8 a strongly hyperbolic unit disk graph with radius R can be covered by $\max\{2\pi\sqrt{2}, 2\pi e^{R/2}\}$ cliques. In particular, for $R < 1$, this yields a bound of $2\pi\sqrt{e}$. Since processing each root removes at least one such clique from the graph, the number of roots in the

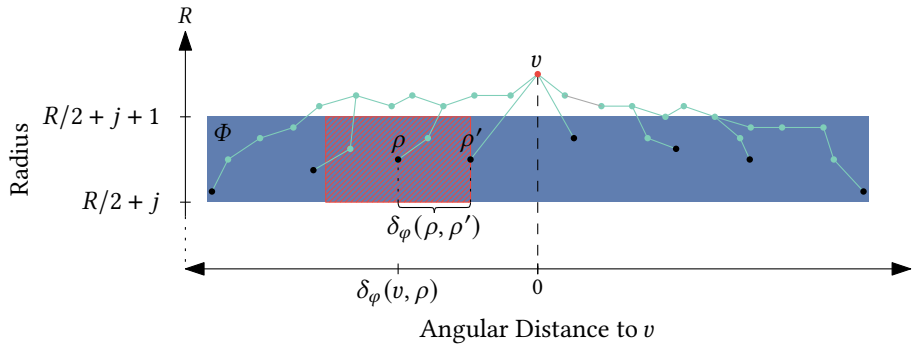


Figure 4.3: Illustration of the proof of Lemma 4.10. A vertex v (red dot) and the roots (black vertices) that are contained in the jth band and are connected to v (via the green paths). All roots lie in the angular interval Φ (blue region). Other than ρ, no root can lie in the red region.

phase is bounded by the number of cliques. It follows, considering the first clique in $\mathcal{D}_{R/2}$ and the remaining cliques when $R < 1$, that we can bound the roots of v in phase i as $|\boldsymbol{\rho}_i(v)| \leq 1 + 2\pi\sqrt{e} \leq 2\pi e$, which we account for with the +2 in the lemma statement.

For the remaining roots of v we can now assume that $R \geq 1$ and that all vertices have radius at least $R/2$. Furthermore, it suffices to show that there are at most $\pi e R(1+a)b^i$ such roots. We cover the remainder of the disk with $R/2$ bands of radial width 1, where the jth band (for $j \in \{0, \ldots, R/2-1\}$) contains all points with radius in $[R/2+j, R/2+j+1]$, see Figure 4.2. The claim then follows if we can bound the number of roots of v in a single band by $2\pi e(1+a)b^i$.

Let $\boldsymbol{\rho}_{i,j}(v)$ denote the roots of v that lie in the jth band (see Figure 4.3). We first bound the angular distance between v and a root in $\boldsymbol{\rho}_{i,j}(v)$, and with that the width of the angular interval Φ that contains all of them (blue region in Figure 4.3). Afterwards, we show that each root reserves a portion of Φ that no other root can be in. An upper bound on $|\boldsymbol{\rho}_{i,j}(v)|$ is then obtained by the quotient of the widths of the two intervals.

Consider a root $\rho \in \boldsymbol{\rho}_{i,j}(v)$. Since the roots are processed in order of increasing radius, all vertices of radius at most $r(\rho)$ have been removed before. Consequently, the path from ρ to v in the partial shortest-path tree rooted at ρ consists only of vertices with radius at least $r(\rho)$. Moreover, in phase i the depth of this tree is $(1+a)b^i$, which means that the path between ρ and v is at most

this long. Therefore, we can apply Lemma 4.8 to conclude that the maximum angular distance between v and a root ρ is at most

$$\begin{aligned}\max_{\rho\in\boldsymbol{\rho}_{i,j}(v)}\delta_\varphi(v,\rho) &\le \max_{\rho\in\boldsymbol{\rho}_{i,j}(v)}(1+a)b^i\cdot\pi e^{R/2-r(\rho)}\\ &\le (1+a)b^i\cdot\pi e^{-j},\end{aligned}$$

where the second inequality stems from the fact that $r(\rho)\ge R/2+j$ holds for all $\rho\in\boldsymbol{\rho}_{i,j}(v)$. Moreover, since roots cannot be adjacent (as they would otherwise remove each other) and all roots in $\boldsymbol{\rho}_{i,j}(v)$ have their radii in $[R/2+j, R/2+j+1]$, we can apply Lemma 4.9 to conclude that the minimum angular distance between two roots $\rho,\rho'\in\boldsymbol{\rho}_{i,j}(v)$ is at least

$$\begin{aligned}\min_{\rho\neq\rho'\in\boldsymbol{\rho}_{i,j}(v)}\delta_\varphi(\rho,\rho') &\ge e^{R/2-(R/2+j+1)}\\ &= e^{-(j+1)}.\end{aligned}$$

Note that the angular interval Φ extends to both angular directions from v. It follows that

$$\begin{aligned}|\boldsymbol{\rho}_{i,j}(v)| &\le 2\cdot\frac{\max_{\rho\in\boldsymbol{\rho}_{i,j}(v)}\delta_\varphi(v,\rho)}{\min_{\rho\neq\rho'\in\boldsymbol{\rho}_{i,j}(v)}\delta_\varphi(\rho,\rho')}\\ &\le \frac{2(1+a)b^i\cdot\pi e^{-j}}{e^{-(j+1)}}\\ &= 2\pi e(1+a)b^i.\end{aligned}$$

□

Recall that the radius of the shortest-path trees that are removed from the graph in two consecutive phases increases by a factor of b and that the algorithm terminates when the first tree in a phase deletes the whole graph. It follows that there are at most $\lceil\log_b(\operatorname{diam}(G))\rceil$ phases. By summing over all phases, we can complete the proof of Theorem 4.7, which we restate for the sake of readability.

Theorem 4.7. *Let G be a strongly hyperbolic unit disk graph with radius $R>0$. Given the disk representation of G, $a>0$, and $b>1$, the* Proton *algorithm with the radially increasing root selection strategy computes a (c,d,k)-tree-cover of G with $c=1+2b/a$, $d=2$, and*

$$k=\pi e\left(\frac{1+a}{b-1}(b^2\operatorname{diam}(G)-1)R+2\big(\log_b(\operatorname{diam}(G))+2\big)\right).$$

Proof. First note that the values for c and d hold for any graph due to Lemma 4.6. It remains to show that the stated bound on k is valid. To that end, we make use of Lemma 4.10, which states that v is contained in at most $\pi e(R(1+a)b^i + 2)$ trees in phase i, and sum over all phases. Since the radius of the shortest-path trees that are removed from the graph in two consecutive phases increases by a factor of b and the algorithm terminates when the first tree in a phase deletes the whole graph, there are at most $\lceil \log_b(\operatorname{diam}(G)) \rceil$ phases. Thus,

$$\begin{aligned} k &= \sum_{i=0}^{\log_b(\operatorname{diam}(G))+1} \pi e(R(1+a)b^i + 2) \\ &= \pi e\left(R(1+a)\left(\sum_{i=0}^{\log_b(\operatorname{diam}(G))+1} b^i\right) + 2\big(\log_b(\operatorname{diam}(G)) + 2\big)\right). \end{aligned}$$

Note that the remaining sum is a partial sum of a geometric series with $b > 1$, which can be computed as $\sum_{i=0}^{\ell} b^i = (b^{\ell+1} - 1)/(b-1)$. We obtain

$$\begin{aligned} k &= \pi e\left(R(1+a)\frac{b^{\log_b(\operatorname{diam}(G))+2} - 1}{b-1} + 2\big(\log_b(\operatorname{diam}(G)) + 2\big)\right) \\ &= \pi e\left(\frac{1+a}{b-1}(b^2 \operatorname{diam}(G) - 1)R + 2\big(\log_b(\operatorname{diam}(G)) + 2\big)\right). \end{aligned}$$

□

While this result holds for all strongly hyperbolic unit disk graphs, more specific results can be obtained when looking at hyperbolic random graphs, which are strongly hyperbolic unit disk graphs where vertices are drawn independently and uniformly at random from a disk of radius $R = \mathrm{O}\big(\log(n)\big)$, see Section 3.3. Since these graphs have a diameter of $\mathrm{O}\big(\log(n)\big)$ asymptotically almost surely [MS19], we obtain the following corollary.

Corollary 4.11. *Let G be a hyperbolic random graph. Given the disk representation of G, $a > 0$, and $b > 1$, the* Proton *algorithm with the radially increasing root selection strategy computes a (c, d, k)-tree-cover of G with $c = 1 + 2b/a$, $d = 2$, and, asymptotically almost surely*

$$k = \mathrm{O}\left(\frac{(1+a)b^2}{b-1} \cdot \log(n)^2\right).$$

Together with Theorem 4.4, it follows that greedy routing on hyperbolic random graphs can be implemented such that the resulting scheme is starvation-free and has stretch $1 + 2b/a$ with additive bound 2. Moreover, by setting $a = b = 2$ we obtain a multiplicative stretch of 3, and can derive that the scheme, asymptotically almost surely, stores $O(\log(n)^4)$ bits at each vertex and takes $O(\log(n)^2)$ time per query.

4.6 Empirical Evaluation

Finally, to evaluate how well our results translate to real-world networks, we performed experiments on 50 graphs from the Network Data Repository [RA15], with sizes ranging from 14 k to over 2.3 M vertices. Since we do not have unit disk representations for these, we used the degrees of the vertices as a proxy for their place in the hierarchical structure. That is, the root selection strategy processed the vertices by decreasing degree. For each graph, we computed a tree-cover using the Proton algorithm with parameters $a = b = 2$, and sampled 10 k vertex pairs for which the path obtained by our routing scheme was compared to a shortest path between them. Figure 4.4 shows boxplots aggregating our observations.

As expected, the maximum observed stretch is 3. However, this stretch occurred only rarely. In all networks most of the sampled routes had a stretch of at most 1.5 and in 16 of the 50 graphs the median stretch was 1. At the same time, the number of trees that a vertex was contained in on average remained small. In 42 of the 50 networks this number was less than 50, even in networks with over 2.3 M vertices.

Figure 4.4: Multiplicative stretch when routing with a tree-cover obtained using the Proton algorithm with $a = b = 2$. Colors show the number of trees k that an average vertex is contained in. The boxes denote the interquartile range extending to the 25th and 75th percentile with horizontal bars showing the median. Whiskers extend to 0.1% and 99.9%, while circles show values beyond that.

5 Bidirectional BFS in Hyperbolic Random Graphs

This chapter is based on joint work with Thomas Bläsius, Cedric Freiberger, Tobias Friedrich, Felix Montenegro-Retana, and Marianne Thieffry [Blä+18; Blä+22]. It started as a student project where we developed some of the core ideas of the paper, yielding a bound of the size of the search space of the bidirectional breadth-first search on hyperbolic random graphs in terms of the number of visited vertices. This was later extended to a bound on the running time by also considering the visited edges, while making the probabilistic statements stronger, and by adding a comparison to the performance of the bidirectional breadth-first search on random graphs with an underlying Euclidean geometry.

5.1 Introduction

In this chapter, we consider another path-finding algorithm. In contrast to the previous chapter, we are now interested in finding an actual shortest path between two vertices, instead of an approximation of it. Besides being of independent interest, many algorithms use shortest path queries as a subroutine. On unweighted graphs, such queries can be answered in linear time using a breadth-first search (BFS). Though this is optimal in the worst case, it is not efficient enough when dealing with large networks or problems involving many shortest path queries.

A way to heuristically improve the run time, is to use a *bidirectional BFS* [Poh69]. It runs two searches, simultaneously exploring the graph from the start and the destination. The shortest path is found once the two search spaces touch. Being one of the standard heuristics, the bidirectional BFS is widely used in practice (e.g., in route planning). On homogeneous networks (where most vertices have similar degrees, like road networks) this typically leads to a speedup factor of about two. However, on heterogeneous networks, where several approaches to computing shortest paths have been studied extensively [ASK12; LFC17; Pen+12], experiments indicate that the bidirectional BFS yields an asymptotic running time improvement [BN16].

	Homogeneous	Heterogeneous
Independent Edges	Bounded Variance $O(n^{\frac{1}{2}+\epsilon})$ [BN16]	Unbounded Variance $O(n^{\frac{4-\beta}{2}+\epsilon})$ [BN16]
Underlying Geometry	Euclidean Random Graphs $\Theta(n)$ [Folklore]	Hyperbolic Random Graphs $\tilde{O}(n^{2\frac{\beta-2}{\beta-1}} + n^{\frac{1}{\beta-1}})$ [This chapter]

Table 5.1: Probabilistic bounds on the running time of the bidirectional BFS obtained by analyzing different random graph models. The models (and associated results) are arranged by the heterogeneity of the corresponding degree distributions of the graphs and the (in)dependence of edges. Here, n denotes the number of vertices in the graph and ϵ is an arbitrarily small constant. The parameter $\beta \in (2, 3)$ denotes the power-law exponent of the degree distribution in the considered heterogeneous networks.

Despite being such a fundamental heuristic, theory fails at predicting and explaining the observed behavior. The theoretical worst-case running time overshoots the observations by a lot. A more promising approach to explain the performance of the bidirectional BFS is an average-case analysis [BN16; LR89]. There, the idea is to consider instances that are drawn from certain probability distributions instead of assuming the worst case. The results of Borassi and Natale are summarized in the first row of Table 5.1 [BN16]. Their analysis covers a variety of random graph models. On the one hand these include homogeneous networks where the degree distribution has bounded variance, e.g. Erdős-Rényi random graphs. On the other hand, they also consider heterogeneous networks where the variance of the degree distribution is unbounded, e.g. Chung-Lu random graphs with power-law exponent $\beta \in (2, 3)$. However, the results, again, do not match what is observed in practice, as it predicts shorter running times on homogeneous networks than on heterogeneous ones.

The fundamental obstacle that prevents the average-case analysis from producing convincing explanations is that the considered random graph models are not realistic. They assume that edges in the graph are independent of each other, leading to low clustering. However, as explained in Chapter 1, real-world networks typically exhibit locality, i.e., edges in an evolving network tend to form between vertices that are already close in the network.

We resolve this discrepancy by modeling edge dependencies using geometry and extend the comparison in Table 5.1 by adding the second row. More precisely, in this chapter, we analyze the bidirectional BFS on random graph models with underlying geometry. We prove that, with high probability, the bidirectional BFS has a sublinear worst-case running time on the heterogeneous networks generated by the hyperbolic random graph model (see Section 3.3). Additionally, it is not hard to see why there is no asymptotic speedup on the homogeneous networks generated by the Euclidean random graph model. Both results match previous empirical observations. Finally, we interpret these insights and discuss how the heterogeneity of the degree distribution and an underlying geometry affect the running time of the bidirectional breadth-first search.

5.2 Bidirectional Breadth-First Search

In an unweighted and undirected graph $G = (V, E)$, a BFS finds the shortest path between two vertices $s, t \in V$ by starting at s and exploring the graph in levels, where the ith level L_i^s contains the vertices with distance i to s. More formally, the BFS starts with the set $L_0^s = \{s\}$ on level 0. Assuming the levels $L_0^s, \dots, L_i^s$ have been computed already, one obtains the next level L_{i+1}^s as the set of neighbors of vertices in level L_i^s that are not contained in earlier levels. Computing L_{i+1}^s from L_i^s is called an *exploration step*, obtained by *exploring the edges* between vertices in L_i^s and L_{i+1}^s.

The bidirectional BFS runs two BFSs simultaneously. The *forward search* starts at s and the *backward search* starts at t. The shortest path between the two vertices can then be obtained, once the search spaces of the forward and backward search touch. Since the two searches cannot actually be run simultaneously, they alternate depending on their progress. When exactly the two searches alternate is determined by the *alternation strategy*. We note that we only swap after full exploration steps, i.e., we never explore only half of level i of one search before continuing with the other. This has the advantage that we can be certain to know the shortest path once a vertex is found by both searches.

In the following we define the *greedy alternation strategy* as introduced by Borassi and Natale [BN16] and show that it is not much worse than any other alternation strategy. Assume the latest levels of the forward and backward searches are L_i^s and L_j^t, respectively. Then the next exploration step of the

forward and backward search would cost time proportional to

$$c_i^s = \sum_{v \in L_i^s} \deg(v), \quad \text{and} \quad c_j^t = \sum_{v \in L_j^t} \deg(v),$$

respectively. The *greedy alternation strategy* then greedily continues with the search that causes the fewer cost in the next exploration step, i.e., it continues with the forward search if $c_i^s \leq c_j^t$ and with the backward search otherwise.

Theorem 5.1. *Let G be a graph with diameter d. If there exists an alternation strategy such that the bidirectional BFS explores $f(n)$ edges, then the bidirectional BFS with greedy alternation strategy explores at most $d \cdot f(n)$ edges.*

Proof. Let A be the alternation strategy that explores only $f(n)$ edges. First note that the number of explored edges only depends on the number of levels explored by the two different searches and not on the actual order in which they are explored. Thus, if the greedy alternation strategy is different from A, we can assume without loss of generality that the greedy strategy performed more exploration steps in the forward search and fewer in the backward search compared to A. Let c^s and c^t be the number of edges explored by the forward and backward search, respectively, when using the greedy strategy. Moreover, let j be the last level of the backward search (which is actually not explored) and, accordingly, let c_j^t be the number of edges the next step in the backward search would have explored. Then $c^t + c_j^t \leq f(n)$ as, when using A, the backward search still explores level j. Moreover, the forward search with the greedy strategy explores at most $c^t + c_j^t$ (and therefore at most $f(n)$) edges in each step, as exploring the backward search would be cheaper otherwise. Consequently, each step in the forward and backward search costs at most $f(n)$. As there are at most d steps in total, we obtain the claimed bound. □

5.2.1 Euclidean Random Graphs

Recall that, Euclidean random graphs, i.e, random Euclidean unit disk graphs, are generated by distributing n vertices uniformly at random in the unit square $[0, 1]^2$ and connecting any two vertices if the Euclidean distance between them is at most some threshold $R \in \mathbb{R}$, as explained in Chapter 1. One can imagine, that each vertex is equipped with a disk of radius R and an edge is added to all other vertices that lie in this disk. The threshold R affects the properties of

the generated network and in order to obtain graphs with a giant component of linear size (as is the case for hyperbolic random graphs), R has to be chosen from the so called *supercritical regime* [Pen03]. In contrast to hyperbolic random graphs, the uniform sampling of the vertices in the Euclidean space leads to a distribution where the number of vertices falling into each disk is roughly the same, which in turn leads to a homogeneous degree distribution.

We examine how a BFS explores such a graph, by considering the region of the plane containing the vertices visited after several exploration steps. When R is chosen from the supercritical regime, then for two vertices at graph theoretic distance d it holds that $R \cdot d$ is at most a constant factor larger than the Euclidean distance between them, if d is super-logarithmic [FSS13]. Additionally, it is easy to see that the Euclidean distance between them can be at most $R \cdot d$. Therefore, we can assume that after k (sufficiently many) steps the region in the plane that contains the visited vertices resembles a disk of radius proportional to k. Since the area of a disk with radius r grows as πr^2, the expected number of explored vertices is in $\Theta(nk^2)$, as the vertices are distributed uniformly.

In this scenario it is easy to see that the performance of a bidirectional BFS improves by a constant factor, compared to a standard BFS. Let s and t be two vertices with (sufficiently large) graph theoretic distance d from each other. Then, the expected number of vertices explored by a standard BFS from s to t is $\Theta(nd^2)$. If we run two searches instead (one starting at s, the other at t), then the expected explored search space is minimized when the two BFSs touch after half as many steps, exploring two disks of half the radius. (Note that this holds independent of the chosen alternation strategy.) In that case the expected number of explored vertices is proportional to $2n\pi(d/2)^2$ which is again $\Theta(nd^2)$, indicating that the bidirectional variant yields no asymptotic speedup over the standard BFS.

In the following, we focus on the performance of the bidirectional BFS on hyperbolic random graphs. They feature a heterogeneous degree distribution, leading to significant differences in the performance of the bidirectional BFS.

5.2.2 Hyperbolic Random Graphs

In this section, we analyze the running time of the bidirectional BFS and obtain an upper bound on the maximum running time over all possible start–destination pairs. Our results are summarized in the following main theorem, where the bound is expressed as a function of the number of vertices n in the graph, as well its maximum degree, denoted by $\deg_{\max}(G)$.

Theorem 5.2. *Let G be a hyperbolic random graph with n vertices. With high probability the shortest path between any two vertices in G can be computed in time $\tilde{O}\big(n^{2-1/\alpha} + n^{1/(2\alpha)} + \deg_{\max}(G)\big)$.*

Note that this bound on the running time also holds in expectation. Our bound fails with probability $O(1/n)$, in which case the worst-case running time is still bounded by the size of the hyperbolic random graph, which is $O(n)$. Consequently, this case contributes $O(1)$ to the expectation, which is dominated by the above bound.

Recall that for the relevant regime of the parameter $\alpha \in (1/2, 1)$, hyperbolic random graphs contain a giant component, while all other components have polylogarithmic size, with high probability (see Section 3.3). Since a bidirectional BFS could completely explore a non-giant component in $\tilde{O}(1)$ time and either return the shortest path (if both vertices are in the same non-giant component) or conclude that the vertices are in different components, we focus on the non-trivial case in the following, where the two considered vertices are both in the giant component.

To prove Theorem 5.2, we make use of the hyperbolic geometry by bounding the numbers of vertices and edges that lie in certain regions within a hyperbolic disk. See Figure 5.1 for an illustration of such regions. In particular, they include *sectors*, i.e., restrictions of a disk $\mathcal{D}_r$ to all points whose angular coordinates lie in a certain interval. The *angular width* of a sector is the length of this interval. For an arbitrary set of points $\mathcal{A}$, we use $\mathcal{A}|_{r_1}^{r_2}$ to denote the restriction of $\mathcal{A}$ to points with radii in $[r_1, r_2]$, i.e., $\mathcal{A}|_{r_1}^{r_2} = \mathcal{A} \cap (\mathcal{D}_{r_2} \setminus \mathcal{D}_{r_1})$.

The general idea for the proof of Theorem 5.2 now is as follows; see Figure 5.2. As long as the two searches visit only low-degree vertices, all explored vertices lie within a small region, i.e., the searches operate locally. Once the searches visit high-degree vertices closer to the center of the hyperbolic disk (green area in Figure 5.2), it takes only few steps to complete the search, as hyperbolic random graphs have a densely connected core. Thus, we split our analysis in two phases: a first phase in which both searches advance towards the center and a second phase in which both searches meet in the center. Note that this strategy assumes that we know the coordinates of the vertices as we would like to stop a search once it reached the center. However, as shown above, the greedy alternation strategy is oblivious to the geometry but performs not much worse than any other alternation strategy.

We can capture the two phases geometrically by separating the whole disk $\mathcal{D}_R$

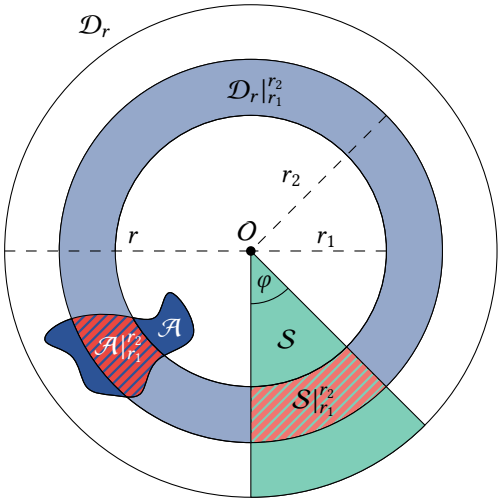


Figure 5.1: Geometric shapes and their intersections. Sector $\mathcal{S}$ (green) has an angular width of φ. Red shaded areas are restricted to points with radii in $[r_1, r_2]$.

into two parts. One is the *inner disk* $\mathcal{D}_R|_0^\rho$ centered at the origin. Its radius ρ is chosen in such a way that any two vertices in $\mathcal{D}_R|_0^\rho$ have a common neighbor with high probability. The second part is the annulus $\mathcal{D}_R|_\rho^R$, the remainder of the whole disk, which we call *outer band*. A single BFS now explores the graph in two phases. In the first phase, the BFS explores vertices in the outer band. The phase ends, when the next vertex to be encountered lies in the inner disk. Once both BFSs completed the first phase, they only need at most two more steps for their search spaces to share a vertex. One step to encounter the vertex[4] in the inner disk and another step to meet at their common neighbor that any two vertices in the inner disk have with high probability; see Figure 5.2.

Note that this scenario describes the worst case. Depending on the positions of the two considered vertices the two searches may touch earlier, e.g., when both vertices are close to each other in the outer band or when at least one of them is already contained in the inner disk. However, since we want to determine an upper bound on the running time, we consider the case where both vertices lie in the outer band and the two searches touch in the inner disk. In the remainder

4 We note that this vertex has a degree of $\tilde{\Omega}(n^{1-1/(2\alpha)})$ with high probability. Consequently, a non-giant component of size $\tilde{O}(1)$ is detected (at the latest) before exploring this vertex.

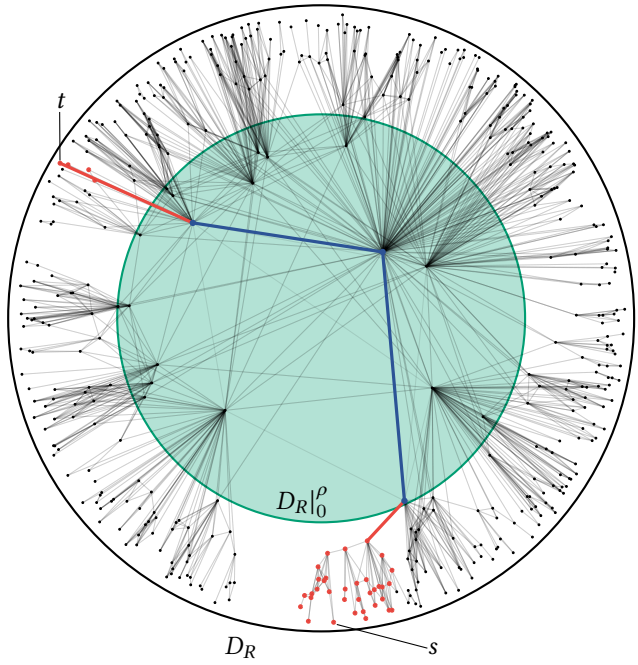


Figure 5.2: Visualization of the two phases of each BFS in a hyperbolic random graph. Vertices visited during the first phase are red. The red edges denote the first encounter of a vertex in the inner disk $\mathcal{D}_R|_0^\rho$ (green region). This corresponds to the first step in the second phase. The last step then leads to a common neighbor via the blue edges.

of the chapter we only consider how one of the two searches explores the graph. The obtained bounds also hold for the other search, meaning the total search space increases only by a constant factor when considering both searches instead of only one.

For our analysis we assume an alternation strategy in which each search stops once it explored one additional level after finding the first vertex in the inner disk $\mathcal{D}_R|_0^\rho$. Of course, this cannot be implemented without knowing the underlying geometry of the network. However, by Theorem 5.1 the search space explored using the greedy alternation strategy is only a polylogarithmic factor larger, as the diameter of hyperbolic random graphs is polylogarithmic with high probability [FK15a]. The following lemma shows for which choice of ρ the above sketched strategy works.

Lemma 5.3. *Let G be a hyperbolic random graph with n vertices. With high probability, G contains a vertex that is adjacent to every other vertex in $\mathcal{D}_R|_0^\rho$, for $\rho = 1/\alpha(\log n - \log\log n)$.*

Proof. Consider a vertex $v \in \mathcal{D}_{R-\rho}$. Note that the distance between two points is upper bounded by the sum of their radii. Thus, every vertex in $\mathcal{D}_R|_0^\rho$ has distance at most R to v, and is therefore adjacent to v. Hence, to prove the claim, it suffices to show the existence of this vertex v with radius at most $R - \rho$. As described in Section 3.3.3, the probability for a single vertex to have radius at most $R - \rho$ is given by the measure $\mu(\mathcal{D}_R|_0^{R-\rho})$. Using Equation (3.8) we obtain

$$\begin{aligned}\mu\left(\mathcal{D}_R|_0^{R-\rho}\right) &= e^{-\alpha\rho}(1+\mathrm{o}(1)) \\ &= \frac{\log n}{n}(1+\mathrm{o}(1)).\end{aligned}$$

Thus, the probability that none of the n vertices lies in $\mathcal{D}_R|_0^{R-\rho}$ is given by

$$\begin{aligned}\Pr\left[V\left(\mathcal{D}_R|_0^{R-\rho}\right) = \emptyset\right] &= \left(1 - \mu\left(\mathcal{D}_R|_0^{R-\rho}\right)\right)^n \\ &= \left(1 - \frac{\log n}{n}(1+\mathrm{o}(1))\right)^n.\end{aligned}$$

Since $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$ (Lemma 2.9), this term can be bounded by

$$\begin{aligned}\Pr\left[V\left(\mathcal{D}_R|_0^{R-\rho}\right) = \emptyset\right] &\leq e^{-\frac{\log(n)}{n}(1+\mathrm{o}(1))\cdot n} \\ &= e^{-\log(n)(1+\mathrm{o}(1))} \\ &= n^{-(1+\mathrm{o}(1))} \\ &= \mathrm{O}\big(1/n\big).\end{aligned}$$

Consequently, there is at least one vertex in $\mathcal{D}_R|_0^{R-\rho}$ with high probability. □

In the following, we first bound the search space explored in the first phase, i.e., before we enter the inner disk $\mathcal{D}_R|_0^\rho$. Afterwards we bound the search space explored in the second phase, which consists of two exploration steps. The first one to enter $\mathcal{D}_R|_0^\rho$ and the second one to find a common neighbor, which exists due to Lemma 5.3.

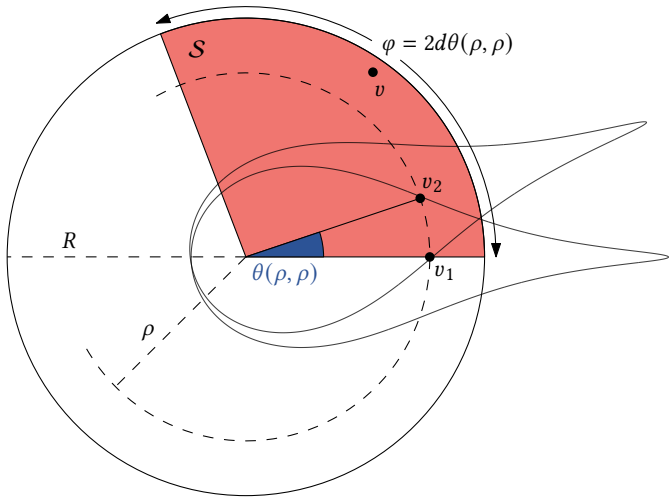


Figure 5.3: The sector $\mathcal{S}$ (red) of angular width φ contains the search space of a BFS starting at v, in the outer band $\mathcal{D}_R|_\rho^R$. The vertices v_1 and v_2 are at maximum angular distance to still be adjacent.

Search Space in the First Phase

To bound the size of the search space in the outer band, we make use of the geometry in the following way. For two vertices in the outer band to be adjacent, their angular distance has to be small. Moreover, the number of exploration steps is bounded by the diameter of the graph. Thus, the maximum angular distance between vertices visited in the first phase cannot be too large. Note that the following lemma restricts the search to a sublinear portion of the disk, which we later use to show that also the number of explored edges is sublinear.

Lemma 5.4. *With high probability, all vertices that a BFS on a hyperbolic random graph explores before finding a vertex with radius at most* $\rho = 1/\alpha(\log n - \log\log n)$ *lie within a sector of angular width* $\tilde{\mathrm{O}}\big(n^{-(1/\alpha-1)}\big)$.

Proof. For an illustration of the proof see Figure 5.3. Recall that $\theta(r_1, r_2)$ denotes the maximum angular distance between two vertices of radii r_1 and r_2 such that they are still adjacent, as described in Section 3.3.5. Since r_1 and r_2 only appear as negative exponents in the expression for $\theta(r_1, r_2)$ (see Equation (3.10)), this angle increases with decreasing radii. Thus, $\theta(r_1, r_2) \leq \theta(\rho, \rho)$ holds for all vertices in the outer band $\mathcal{D}_R|_\rho^R$.

Now assume we start a BFS at a vertex $v \in \mathcal{D}_R|_\rho^R$ and perform d exploration steps without leaving the outer band $\mathcal{D}_R|_\rho^R$. Then no explored vertex has angular distance more than $d\theta(\rho,\rho)$ from v. Thus, the whole search space lies within a disk sector of angular width $2d\theta(\rho,\rho)$. The number of steps d is at most polylogarithmic as the diameter of a hyperbolic random graph is polylogarithmic with high probability [FK15a]. Using Equation (3.10) for $\theta(\rho,\rho)$, we obtain

$$\begin{aligned}\theta(\rho,\rho) &= 2e^{\frac{R-2\rho}{2}}\left(1 \pm \Theta\left(e^{R-2\rho}\right)\right)\\ &= 2e^{C/2}n^{1-1/\alpha}\log(n)^{1/\alpha}\left(1+\Theta\left(\left(\frac{\log n}{n^{1-\alpha}}\right)^{2/\alpha}\right)\right)\\ &= \mathrm{O}\left(n^{-(1/\alpha-1)}\log(n)^{1/\alpha}\right),\end{aligned}$$

which proves the claimed bound. □

Note that the expected number of vertices in a sector $\mathcal{S}$ of angular width φ is linear in $n\varphi$ due to the fact that the angular coordinate of each vertex is chosen uniformly at random. Thus, Lemma 5.4 already shows that the expected number of vertices visited in the first phase of the BFS is $\tilde{\mathrm{O}}\left(n^{2-1/\alpha}\right)$, which is sublinear in n. It is also not hard to see that this bound holds with high probability (see Corollary 2.3). To also bound the number of explored edges, we sum the degrees of vertices in $\mathcal{S}$. It is not surprising that this yields the same asymptotic bound in expectation, as the average degree in a hyperbolic random graph is constant, with high probability (again, see Section 3.3). However, to obtain meaningful results, we need a bound that holds with high probability. Though we can use techniques similar to those that have been used to show that the average degree of the whole graph is constant [GPP12; Keu18], the situation is complicated by the restriction to a sublinear portion of the disk. Nonetheless, we obtain the following theorem.

Theorem 5.5. *Let G be a hyperbolic random graph with n vertices. Then, the degrees of vertices in every sector of angular width $\varphi = \Omega\left(\log(n)^2/n^{1/2}\right)$ sum to $\tilde{\mathrm{O}}\left(\varphi n + n^{1/(2\alpha)} + \deg_{\max}(G)\right)$ with high probability.*

We note that $\deg_{\max}(G)$ has to be included here, as the theorem states a bound for every sector, and thus in particular for sectors containing the vertex of maximum degree. Recall, that $\deg_{\max}(G) = \tilde{\mathrm{O}}\left(n^{1/(2\alpha)}\right)$ holds almost surely [GPP12],

and that there exists a bound of $O(n^{1/(2\alpha-\varepsilon)})$ for all $\varepsilon > 0$ that holds with high probability [Keu18]. Moreover, we note that the condition $\varphi = \Omega(\log(n)^2/n^{1/2})$ is crucial for our proof, i.e., the angular width of the sector has to be sufficiently large for the concentration bound to hold. Also note that, depending on α, the angular width determined in Lemma 5.4 may be smaller than this lower bound. However, if this is the case, we can choose $\varphi = \tilde{O}(n^{-1/2})$ as an upper bound for the angular width of the sector and obtain $\tilde{O}(\varphi n) = \tilde{O}(n^{1/2}) = \tilde{O}(n^{1/(2\alpha)})$ for $\alpha \in (1/2, 1)$. Consequently, the bound holds for the previously determined angular width $\tilde{O}(n^{-(1/\alpha-1)})$ for all $\alpha \in (1/2, 1)$.

As the proof for Theorem 5.5 is rather technical, we defer it to Section 5.3. Together with Lemma 5.4, we obtain the following corollary. Note that since $\alpha \in (1/2, 1)$, this shows that the running time spend in the first phase (not accounting for the maximum degree) is sublinear in n with high probability.

Corollary 5.6. *On a hyperbolic random graph the first phase of the bidirectional search explores* $\tilde{O}(n^{2-1/\alpha} + n^{1/(2\alpha)} + \deg_{\max}(G))$ *edges with high probability.*

Search Space in the Second Phase

The first phase of the BFS is completed when the next vertex to be encountered lies in the inner disk. Thus, the second phase consists of only two exploration steps. One step to encounter the vertex in the inner disk and another step to meet the other search. Thus, to bound the running time of the second phase, we have to bound the number of edges explored in these two exploration steps. To do this, let V_1 be the set of vertices encountered in the first phase. Recall that all these vertices lie within a sector $\mathcal{S}$ of angular width $\varphi = \tilde{O}(n^{-(1/\alpha-1)})$, by Lemma 5.4. The number of explored edges in the second phase is then bounded by the sum of degrees of all neighbors $N(V_1)$ of vertices in V_1. To bound this sum, we divide the neighbors of V_1 into two categories: $N(V_1) \cap V(\mathcal{S})$ and $N(V_1) \setminus V(\mathcal{S})$. Note that we already bounded the sum of degrees of vertices in $\mathcal{S}$ for the first phase (see Theorem 5.5), which clearly also bounds this sum for $N(V_1) \cap V(\mathcal{S})$. Thus, it remains to bound the sum of degrees of vertices in $N(V_1) \setminus V(\mathcal{S})$.

To bound this sum, we introduce two *hypothetical vertices* (i.e., vertices with specific positions that are not actually part of the graph) c_1 and c_2 such that every vertex in $N(V_1) \setminus V(\mathcal{S})$ is a neighbor of c_1 or c_2. Then it remains to bound the sum of degrees of neighbors of these two vertices. To define c_1 and c_2, recall that the first phase is restricted to points in the sector $\mathcal{S}$ that have a radius

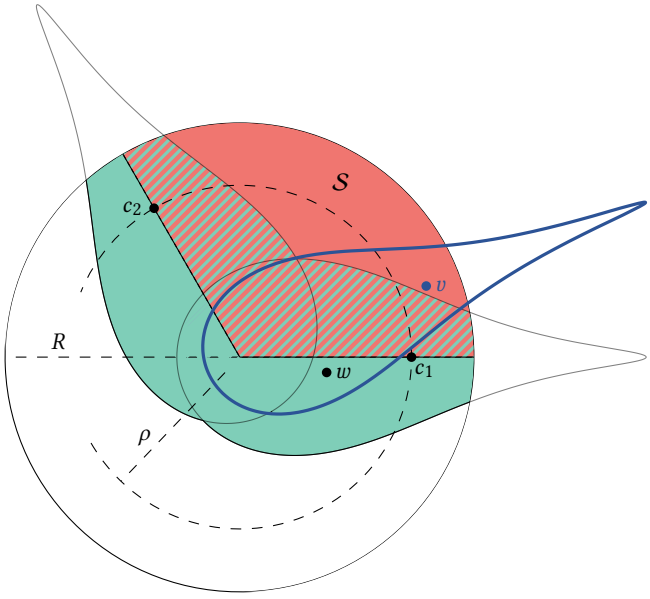


Figure 5.4: Neighbor w of vertex v (blue) is in $\mathcal{S}$ (red) or a neighbor of c_1 or c_2 (green).

greater than ρ, i.e., all vertices in V_1 lie within $\mathcal{S}|_\rho^R$. The hypothetical vertices c_1 and c_2 are basically positioned at the corners of this region, i.e., they both have radius ρ, and they assume the maximum and minimum angular coordinate within $\mathcal{S}$, respectively. Figure 5.4 shows these positions. We obtain the following.

Lemma 5.7. *Let G be a hyperbolic random graph, let $\mathcal{S}$ be a sector, and let $v \in V(\mathcal{S}|_\rho^R)$ be a vertex. Then, every neighbor of v lies in $\mathcal{S}$ or is a neighbor of one of the hypothetical vertices c_1 or c_2.*

Proof. Let $v = (r, \varphi) \in V(\mathcal{S}|_\rho^R)$ and $w \in N(v) \setminus V(\mathcal{S})$ be two vertices. Without loss of generality, assume that c_1 lies between v and w, as is depicted in Figure 5.4. Now consider the point $v' = (\rho, \varphi)$ obtained by moving v to the same radius as c_1. According to Corollary 3.3 we have $N(v) \subseteq N(v')$. In particular, it holds that $w \in N(v')$ and the hyperbolic distance between v' and w is therefore bounded by $\delta_{\mathbb{H}^2}(v', w) \leq R$. Since v' and c_1 have the same radial coordinate and c_1 is between v' and w, meaning $\delta_\varphi(v', c_1) + \delta_\varphi(c_1, w) = \delta_\varphi(v', w)$, it follows that $\delta_{\mathbb{H}^2}(c_1, w) \leq R$. Consequently, w is a neighbor of c_1. □

By the above argument, it remains to sum the degrees of neighbors of c_1 and c_2. In the following, we show that the degrees of the neighbors of a vertex with radius r sum to $\Theta\left(ne^{-(\alpha-1/2)r}\right)$ in expectation. We note that, for large values of r, i.e., for a vertex lying close to the boundary of the disk, this term is surprisingly large. This is due to the fact that, although vertices near the center of the disk are rather unlikely to exist in the first place, their degree would be sufficiently large such that they dominate the expected degree sum.

Lemma 5.8. *Let G be a hyperbolic random graph and let v be a vertex. Then, the degrees of the neighbors of v sum to $\Theta\left(ne^{-(\alpha-1/2)r(v)}\right)$ in expectation.*

Proof. Let Z_v be the sum of the degrees of the neighbors of v, which is a random variable that depends on the positions of all vertices in the graph. Formally, we can express Z_v by assigning each vertex $w \in V \setminus \{v\}$ two random variables X_w and Y_w. The first is an indicator random variable with $X_w = 1$ if w is a neighbor of v and $X_w = 0$ otherwise. Additionally, the random variable Y_w denotes the degree of w. The sum of the degrees of the neighbors of v can then be written as

$$Z_v = \sum_{w \in V \setminus \{v\}} X_w \cdot Y_w.$$

The expected value of Z_v is given by

$$\mathbb{E}[Z_v] = \mathbb{E}\left[\sum_{w \in V \setminus \{v\}} X_w \cdot Y_w\right] = \sum_{w \in V \setminus \{v\}} \mathbb{E}[X_w \cdot Y_w],$$

where the second equality holds due to the linearity of expectation. To compute the expected value of $X_w \cdot Y_w$ we can apply the law of total expectation (see Section 2.2) and obtain

$$\mathbb{E}[Z_v] = \sum_{w \in V \setminus \{v\}} \sum_{x \in \{0,1\}} \mathbb{E}\left[X_w \cdot Y_w \mid X_w = x\right] \cdot \Pr[X_w = x].$$

Clearly, the case where $X_w = 0$ does not contribute anything to the sum, which can thus be simplified as

$$\mathbb{E}[Z_v] = \sum_{w \in V \setminus \{v\}} \mathbb{E}\left[Y_w \mid X_w = 1\right] \cdot \Pr[X_w = 1].$$

Recall that $X_w = 1$ denotes the event where w is a neighbor of v. That is,

$$\Pr[X_w = 1] = \Pr\big[w \in N(v)\big] = \Pr\big[w \in V(\mathcal{D}_R(v))\big] = \mu(\mathcal{D}_R(v)).$$

Moreover, recall that Y_w denotes the random variable representing the degree of w. Consequently, we can now write $\mathbb{E}[Z_v]$ as

$$\begin{aligned}\mathbb{E}[Z_v] &= \sum_{w \in V \setminus \{v\}} \mu(\mathcal{D}_R(v)) \cdot \mathbb{E}\big[Y_w \,\big|\, w \in V(\mathcal{D}_R(v))\big] \\ &= (n-1) \cdot \mu(\mathcal{D}_R(v)) \cdot \mathbb{E}\big[\deg(w) \,\big|\, w \in V(\mathcal{D}_R(v))\big]. \end{aligned} \tag{5.1}$$

We continue by computing the expected degree of a vertex w *conditioned* on the fact that it is contained in $\mathcal{D}_R(v)$. To this end, we first consider the expected value without the condition, analogous to how it was done previously [GPP12, Proof of Theorem 2.3], and afterwards explain how to incorporate the condition. The expected degree of a vertex w with fixed radius r is given by

$$\mathbb{E}\big[\deg(w) \,\big|\, r(w) = r\big] = (n-1)\mu(\mathcal{D}_R(w)).$$

To obtain the expected degree of w without fixing its radius (or angle for that matter) we then integrate $\mathbb{E}\big[\deg(w) \,\big|\, r(w) = r \wedge \varphi(w) = \varphi\big] \cdot f(r, \varphi)$ (note the joint distribution) over the whole disk. That is,

$$\begin{aligned}\mathbb{E}\big[\deg(w)\big] &= \iint_{\mathcal{D}_R} \mathbb{E}\big[\deg(w) \,\big|\, r(w) = r \wedge \varphi(w) = \varphi\big] \cdot f(r, \varphi)\, \mathrm{d}\varphi\, \mathrm{d}r \\ &= \iint_{\mathcal{D}_R} \mathbb{E}\big[\deg(w) \,\big|\, r(w) = r\big] \cdot f(r, \varphi)\, \mathrm{d}\varphi\, \mathrm{d}r, \end{aligned}$$

where the second step follows from the fact that the expected degree of a vertex is independent of its angular coordinate.

It remains to include the condition on the fact that w cannot be anywhere in the whole disk but lies in $\mathcal{D}_R(v)$ instead. First, we have to accommodate for the fact that if w is a neighbor of v, as then conversely v is also a neighbor of w. Consequently, we know that w has at least one neighbor, which we reflect in the expected value by introducing the condition on v lying inside $\mathcal{D}_R(w)$. Moreover, in general the conditional expectation of a random variable X conditioned on an event A (with $\Pr[A] > 0$) is given by $\mathbb{E}\big[X \mid A\big] = \int_{-\infty}^{\infty} x \cdot f_{X|A}(x)\, \mathrm{d}x$, where $f_{X|A}$

is defined as

$$f_{X|A}(x) = \begin{cases} \frac{f_X(x)}{\Pr[A]}, & x \in A, \\ 0, & x \notin A, \end{cases}$$

as explained in Section 2.2. Therefore, the above expression for the expected degree of w can be adjusted to include the condition as

$$\mathbb{E}\big[\deg(w) \,\big|\, w \in V(\mathcal{D}_R(v))\big] = \iint_{\mathcal{D}_R(v)} \mathbb{E}\big[\deg(w) \,\big|\, r(w) = r \wedge v \in V(\mathcal{D}_R(w))\big] \cdot \frac{f(r,\varphi)}{\Pr\big[w \in V(\mathcal{D}_R(v))\big]} \,\mathrm{d}\varphi \,\mathrm{d}r.$$

Note that the probability $\Pr\big[w \in V(\mathcal{D}_R(v))\big]$ in the denominator is, again, the measure $\mu(\mathcal{D}_R(v))$ of the disk of radius R centered at v. Substituting this expression in the above Equation (5.1) for the expected sum $\mathbb{E}[Z_v]$ of the degrees of the neighbors of v, we get

$$\begin{aligned} \mathbb{E}[Z_v] &= (n-1) \cdot \mu(\mathcal{D}_R(v)) \cdot \mathbb{E}\big[\deg(w) \,\big|\, w \in V(\mathcal{D}_R(v))\big] \\ &= (n-1) \cdot \mu(\mathcal{D}_R(v)) \cdot \iint_{\mathcal{D}_R(v)} \mathbb{E}\big[\deg(w) \,\big|\, r(w) = r \wedge v \in V(\mathcal{D}_R(w))\big] \cdot \frac{f(r,\varphi)}{\mu(\mathcal{D}_R(v))} \,\mathrm{d}\varphi \,\mathrm{d}r \\ &= (n-1) \cdot \iint_{\mathcal{D}_R(v)} \mathbb{E}\big[\deg(w) \,\big|\, r(w) = r \wedge v \in V(\mathcal{D}_R(w))\big] \cdot f(r,\varphi) \,\mathrm{d}\varphi \,\mathrm{d}r. \end{aligned}$$

To compute the integral, we determine the expected degree of w conditioned on the fact that $r(w) = r$ and on v being in the neighborhood of w deterministically. Therefore, we obtain the expected degree by adding 1 (for v) to the expected number of vertices among the remaining $V \setminus \{v, w\}$ that are sampled into $\mathcal{D}_R(w)$ and obtain

$$\mathbb{E}\big[\deg(v) \,\big|\, r(w) = r \wedge v \in V(\mathcal{D}_R(w))\big] = 1 + (n-2)\mu(\mathcal{D}_R(w)),$$

which is $1 + \Theta\big(ne^{-r/2}\big)$ due to Equation (3.9). Note that $\Theta\big(ne^{-r/2}\big)$ is $\Omega(1)$ for all

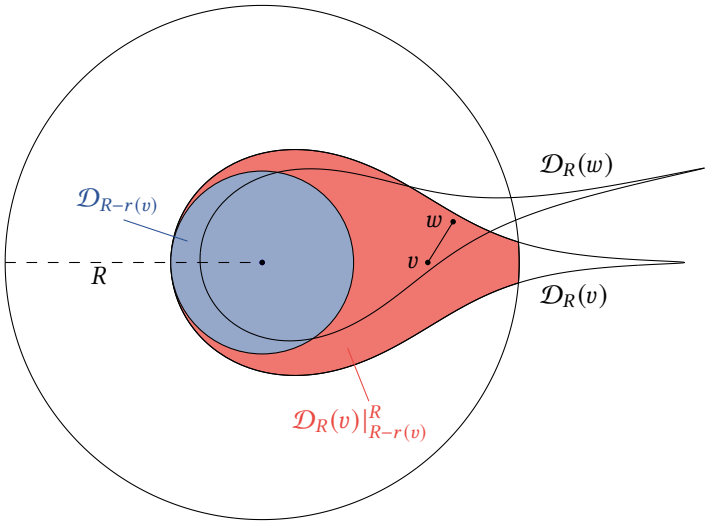


Figure 5.5: Situation in the proof of Lemma 5.8. Vertex w is a neighbor of v. To integrate $\mathcal{D}_R(v) \cap \mathcal{D}_R$, we split the region into two parts: $\mathcal{D}_R(v)|_0^{R-r(v)} = \mathcal{D}_{R-r(v)}$ (blue) and $\mathcal{D}_R(v)|_{R-r(v)}^{R}$ (red).

$r \in [0, R]$, allowing us to further simplify the expected value to

$$\mathbb{E}\left[\deg(v) \;\middle|\; r(w) = r \wedge v \in V(\mathcal{D}_R(w))\right] = \Theta\left(ne^{-r/2}\right).$$

Moreover, recall that $f(r, \varphi) = 0$ for $r > R$ and that it can otherwise be bounded by $f(r, \varphi) = \Theta\left(e^{-\alpha(R-r)}\right)$, see Equation (3.7). We obtain

$$\begin{aligned}\mathbb{E}[Z_v] &= \Theta\left((n-1) \cdot \iint_{\mathcal{D}_R(v) \cap \mathcal{D}_R} ne^{-r/2} \cdot e^{-\alpha(R-r)} \,\mathrm{d}\varphi \,\mathrm{d}r\right) \\ &= \Theta\left(n^2 e^{-\alpha R} \cdot \iint_{\mathcal{D}_R(v) \cap \mathcal{D}_R} e^{(\alpha - 1/2)r} \,\mathrm{d}\varphi \,\mathrm{d}r\right).\end{aligned}$$

We can now split the integral into two parts, where the first contains the disk $\mathcal{D}_R(v)|_0^{R-r(v)} = \mathcal{D}_{R-r(v)}$ and the other contains the remainder of $\mathcal{D}_R(v) \cap \mathcal{D}_R$, which is given by $\mathcal{D}_R(v)|_{R-r(v)}^{R}$ (see Figure 5.5). For the second part we can use Equation (3.10) to bound the angle $\theta(r(v), r)$ up to which we need to integrate

depending on r. As a result, we get

$$\mathbb{E}[Z_v] = \Theta\left(n^2 e^{-\alpha R} \cdot \left(\int_0^{R-r(v)} \int_0^{2\pi} e^{(\alpha-1/2)r} \,\mathrm{d}\varphi \,\mathrm{d}r + \int_{R-r(v)}^{R} \int_0^{\theta(r(v),r)} e^{(\alpha-1/2)r} \,\mathrm{d}\varphi \,\mathrm{d}r \right)\right).$$

Regarding the first part of the sum, note that evaluating the inner integral only contributes a constant factor that can be dropped due the Θ-notation. Computing the outer integral then yields $\Theta\left(e^{(\alpha-1/2)(R-r(v))}\right)$. For the second part of the sum we, again, first evaluate the inner integral and substitute $\theta(r(v), r) = \Theta\left(e^{(R-r(v)-r)/2}\right)$, see Equation (3.10). We obtain

$$\mathbb{E}[Z_v] = \Theta\left(n^2 e^{-\alpha R} \cdot \left(e^{(\alpha-1/2)(R-r(v))} + e^{(R-r(v))/2} \int_{R-r(v)}^{R} e^{-(1-\alpha)r} \,\mathrm{d}r \right)\right).$$

The last integral evaluates to $\mathrm{O}\left(e^{-(1-\alpha)(R-r(v))}\right)$, which multiplied by the factor $e^{(R-r(v))/2}$ yields asymptotically the same expression as the first summand and we get

$$\mathbb{E}[Z_v] = \Theta\left(n^2 e^{-(\alpha-1/2)r(v)} \cdot e^{-R/2}\right).$$

Finally, we can substitute $R = 2\log(n) + C$ in order to obtain the claimed bound $\mathbb{E}[Z_v] = \Theta\left(n e^{-(\alpha-1/2)r(v)}\right)$. □

For c_1 and c_2, which both have radius ρ, the degrees of their neighbors thus sum to $\tilde{\mathrm{O}}\left(n^{1/(2\alpha)}\right)$ in expectation. However, to actually prove Theorem 5.2, we need a bound that holds with high probability for all possible angular coordinates of c_1 and c_2. As with the sum of the degrees in a sector, we prove a slightly weaker bound that matches the one in Theorem 5.2 and holds with high probability. We obtain the following lemma.

Lemma 5.9. *Let G be a hyperbolic random graph and let v be a hypothetical vertex with radius $\rho = 1/\alpha(\log n - \log\log n)$ and arbitrary angular coordinate. Then, the degrees of the neighbors of v sum to $\tilde{\mathrm{O}}\left(n^{2-1/\alpha} + n^{1/(2\alpha)} + \deg_{\max}(G)\right)$ with high probability.*

Again, the proof is rather technical and thus deferred to Section 5.3. Together with the bounds on the sum of degrees in a sector of width $\varphi = \tilde{O}(n^{-(1/\alpha-1)})$ (Theorem 5.5), we obtain the following corollary, which concludes the proof of Theorem 5.2.

Corollary 5.10. *On a hyperbolic random graph the second phase of the bidirectional search explores* $\tilde{O}(n^{2-1/\alpha} + n^{1/(2\alpha)} + \deg_{\max}(G))$ *edges with high probability.*

5.3 Concentration Bounds for the Sum of Vertex Degrees

Here we prove the concentration bounds that were announced in the previous section. For the first phase, we already know that the search space is contained within a sector $\mathcal{S}$ of sublinear width (Lemma 5.4). Thus, the running time in the first phase is bounded by the sum of vertex degrees in this sector. Moreover, all edges explored in the second phase also lie within the same sector $\mathcal{S}$ or are incident to neighbors of the two hypothetical vertices c_1 and c_2 (Lemma 5.7). Thus, the running time of the second phase is bounded by the sum of vertex degrees in $\mathcal{S}$ and in the neighborhood of c_1 and c_2.

In both cases, we have to bound the sum of vertex degrees in certain areas of the disk, which can be done as follows. For each degree, we want to compute the number of vertices of this degree in the considered area and multiply it with the degree. As all vertices with a certain degree have roughly the same radius, we can separate the disk into small annuli, which we call *bands*, one for each degree. Then, summing over all degrees comes down to summing over all bands and multiplying the number of vertices in this band with the corresponding degree. If we can prove that each of these values is highly concentrated, i.e., holds with probability $1 - O(n^{-2})$, we obtain that the sum is concentrated as well (using the union bound, see Section 2.2). Unfortunately, this fails in two situations. For small radii, the number of vertices within the corresponding band (i.e., the number of high degree vertices) is too small to be concentrated. Moreover, for large radii the degree is too small to be concentrated around its expected value.

To overcome this issue, we partition the disk $\mathcal{D}_R$ into three parts. An inner part $\mathcal{D}_R|_0^{\rho_I(\varphi)}$, containing all points of radius at most $\rho_I(\varphi)$, an outer part $\mathcal{D}_R|_{\rho_O}^{R}$, containing all points of radius at least ρ_O, and a central part $\mathcal{D}_R|_{\rho_I(\varphi)}^{\rho_O}$, containing all points in between. We choose $\rho_I(\varphi)$ such that the number of vertices with

maximum degree in a sector part $\mathcal{S}|_{\rho_I(\varphi)}^{\rho_O}$ of angular width φ is $\Omega\big(\log(n)\big)$, which ensures that for each vertex degree, the number of vertices with this degree is concentrated. Moreover, we choose ρ_O in such a way that the vertex degrees in $\mathcal{S}|_{\rho_I(\varphi)}^{\rho_O}$ are sufficiently concentrated. To achieve this, we set

$$\rho_I(\varphi) = R - \frac{1}{\alpha}\left(\log\left(\frac{\varphi}{2\pi}\right) + \log n - \log\log n\right) \text{ and } \rho_O = R - (2+\varepsilon)\log\log(n),$$

for any constant $\varepsilon \in (0,1)$, and show concentration for the sum of the degrees in a sector and in the neighborhood of a vertex with radius ρ, separately for the three parts of the disk.

5.3.1 The Inner Part of the Disk

The inner part $\mathcal{D}_R|_0^{\rho_I(\varphi)}$ contains vertices of high degree. It is not hard to see that there are only logarithmically many vertices with radius at most $\rho_I(\varphi)$.

Lemma 5.11. *Let G be a hyperbolic random graph, let $\varphi \in [0, 2\pi]$ be an angle, and let $\xi > 0$ be a constant. A sector $\mathcal{S}|_0^{\rho_I(\varphi)}$ of angular width $\xi\varphi \in [0, 2\pi]$ contains $\mathrm{O}\big(\log(n)\big)$ vertices, with probability $1 - \mathrm{O}\big(n^{-c}\big)$ for any constant c.*

Proof. By Equation (3.8) the expected number of vertices in $\mathcal{D}_R|_0^{\rho_I(\varphi)}$ is given by

$$\mathbb{E}\Big[\big|V(\mathcal{D}_R|_0^{\rho_I(\varphi)})\big|\Big] = ne^{-\alpha(R-\rho_I(\varphi))}(1+\mathrm{o}(1)).$$

Since the angular coordinates of the vertices are distributed uniformly in $[0, 2\pi)$, the expected number of vertices in a sector portion $\mathcal{S}|_0^{\rho_I(\varphi)}$ of angular width $\xi\varphi$ is

$$\begin{aligned}\mathbb{E}\Big[\big|V(\mathcal{S}|_0^{\rho_I(\varphi)})\big|\Big] &= \frac{\xi\varphi}{2\pi} ne^{-\alpha(R-\rho_I(\varphi))}(1+\mathrm{o}(1)) \\ &= \frac{\xi\varphi}{2\pi} ne^{-(\log(\varphi/2\pi)+\log n-\log\log n)}(1+\mathrm{o}(1)) \\ &= \xi\log(n)(1+\mathrm{o}(1)).\end{aligned}$$

Since $\xi > 0$ is constant, this bound is in $\Omega\big(\log(n)\big)$ and we can apply Corollary 2.3 to conclude that $|V(\mathcal{S}|_0^{\rho_I(\varphi)})| = \mathrm{O}\big(\log(n)\big)$ holds with probability $1 - \mathrm{O}\big(n^{-c}\big)$ for any constant c. □

Note that, if $\varphi = \Omega(1/n)$, we can choose at most $O(n)$ sectors of width 2φ such that any sector of width φ lies completely in one of them. Thus, the probability that there exists a sector portion $\mathcal{S}|_0^{\rho_I(\varphi)}$ where the number of vertices is superlogarithmic, is bounded by the probability that it is too large in at least one of these $O(n)$ sectors (of twice the width). By choosing $\xi = 2$, we can apply Lemma 5.11 to conclude that a single sector $\mathcal{S}|_0^{\rho_I(\varphi)}$ of twice the angular width contains at most $O(\log(n))$ vertices with probability $1 - O(n^{-2})$. After applying the union bound we can bound the number of edges in every such sector portion using the maximum degree $\deg_{\max}(G)$ to obtain the following corollary.

Corollary 5.12. *Let G be a hyperbolic random graph. For every sector $\mathcal{S}$ of angular width $\varphi = \Omega(1/n)$, the degrees of the vertices in $\mathcal{S}|_0^{\rho_I(\varphi)}$ sum to $\tilde{O}(\deg_{\max}(G))$ with high probability.*

Note that, in particular the statement holds for the previously determined angle $\varphi = \tilde{O}(n^{-(1/\alpha-1)})$ for $\alpha \in (1/2, 1)$. Additionally, by setting $\varphi = 2\pi$, we can use Lemma 5.11 to bound the sum of the degrees of the high degree vertices in the neighborhood of a vertex with radius ρ.

Corollary 5.13. *Let G be a hyperbolic random graph. For every vertex v of radius ρ, the degrees of the neighbors of v in $\mathcal{D}_R|_0^{\rho_I(2\pi)}$ sum to $\tilde{O}(\deg_{\max}(G))$ with high probability.*

5.3.2 The Central Part of the Disk

For each possible vertex degree k, we want to compute the number of vertices with this degree in the central part $\mathcal{D}_R|_{\rho_I(\varphi)}^{\rho_O}$. First note, that by Equation (3.9) a vertex with fixed radius has expected degree $\Theta(k)$ if this radius is $2\log(n/k)$. Motivated by this, we define $r_k = 2\log(n/k)$. To bound the sum of degrees in the central part $\mathcal{D}_R|_{\rho_I(\varphi)}^{\rho_O}$, we use that vertices with radius significantly larger than r_k also have a smaller degree. To this end, we first prove that a vertex with degree k can actually not have a radius much larger than r_k. This has the advantage, that we can bound the number of degree-k vertices by bounding the number of vertices with these radii.

Lemma 5.14. *Let G be a hyperbolic random graph. Then, for every constant $c > 0$, there exist constants $\lambda, \tau > 0$, such that all vertices with degree at least $k \geq \lambda \log(n)$ have radius at most $r_k + \tau$ with probability $1 - O(n^{-c})$.*

Proof. To prove this lemma, it suffices to show that there exist constants $\lambda, \tau > 0$, such that the probability of a vertex with radius greater than $r_k + \tau$ having degree at least k, i.e. $\Pr\left[\exists v \in V\colon \deg(v) \geq k \wedge r(v) \geq r_k + \tau\right]$, is small. To obtain the following sequence of inequalities, we first use the union bound, then apply the definition of conditional probabilities, and finally use Corollary 3.3.

$$\begin{aligned}\Pr\left[\exists v \in V\colon \deg(v) \geq k \wedge r(v) \geq r_k + \tau\right] &\leq n \cdot \Pr\left[\deg(v) \geq k \wedge r(v) \geq r_k + \tau\right] \\ &\leq n \cdot \Pr\left[\deg(v) \geq k \,\middle|\, r(v) \geq r_k + \tau\right] \\ &\leq n \cdot \Pr\left[\deg(v) \geq k \,\middle|\, r(v) = r_k + \tau\right].\end{aligned}$$

To prove the lemma, it remains to show that $\Pr\left[\deg(v) \geq k \,\middle|\, r(v) = r_k + \tau\right]$ is sufficiently small, i.e., $\mathrm{O}\left(n^{-(c+1)}\right)$.

Recall that by Equation (3.9), the expected degree of a vertex with radius r is $\Theta\left(ne^{-r/2}\right)$. For a vertex v with radius $r_k + \tau$ we obtain $ne^{-(r_k+\tau)/2} = e^{-\tau/2}k$. It follows that there exists a constant $c' > 0$, such that $\mathbb{E}\left[\deg(v)\right] \leq c'e^{-\tau/2}k$. By choosing τ large enough we can ensure that $k \geq 6\mathbb{E}\left[\deg(v)\right]$, allowing us to apply the Chernoff bound in Theorem 2.1. We obtain $\Pr\left[\deg(v) \geq k\right] \leq 2^{-k}$. Finally, since we have $k \geq \lambda \log(n)$, we can choose λ such that this probability is bounded by $\mathrm{O}\left(n^{-(c+1)}\right)$. □

We are now ready to bound the number of vertices in a sector that have degree at least k. As mentioned earlier, this bound only works for large k as the degree is not sufficiently concentrated otherwise. Moreover, the degree cannot be too large, as otherwise the number of vertices of this degree is not concentrated. The upper bound on k in the following lemma directly corresponds to our choice for $\rho_I(\varphi)$. Additionally, ρ_O is chosen such that the degrees of vertices with radii smaller than ρ_O meet the lower bound on k, i.e., the lemma holds for the central part $\mathcal{S}|_{\rho_I(\varphi)}^{\rho_O}$.

Lemma 5.15. *Let G be a hyperbolic random graph and let $\mathcal{S}$ be a sector of angular width φ. If $k = \omega\left(\log(n)\right)$ and $k = \mathrm{O}\left((\varphi n/\log(n))^{1/(2\alpha)}\right)$, then the number of vertices in $\mathcal{S}$ with degree at least k is $\mathrm{O}\left(\varphi n k^{-2\alpha}\right)$ with probability $1 - \mathrm{O}\left(n^{-c}\right)$ for any constant $c > 0$.*

Proof. By Lemma 5.14 we know that, for any constant $c' > 0$, there are constants $\lambda, \tau > 0$ such that all vertices of degree at least $k \geq \lambda \log(n)$ have radius at most $r_k + \tau$, with probability $1 - \mathrm{O}\left(n^{-c'}\right)$. Since $k = \omega\left(\log(n)\right)$ we have $k \geq \lambda \log(n)$

for large enough n and obtain that, with the same probability, all vertices of degree at least k that are in $\mathcal{S}$ are in $\mathcal{S}|_0^{r_k+\tau}$. Moreover, since the angular width of $\mathcal{S}$ is φ and since the angular coordinates of the vertices are distributed uniformly, the expected number of vertices in $\mathcal{S}|_0^{r_k+\tau}$ is given by $\varphi/(2\pi) \cdot n\mu(\mathcal{D}_R|_0^{r_k+\tau})$. Now we can apply Equation (3.8), which states that a disk of radius $r_k + \tau$ centered at the origin has measure $e^{-\alpha(R-(r_k+\tau))}(1 + o(1))$ and obtain

$$\begin{aligned}\mathbb{E}\left[|V(\mathcal{S}|_0^{r_k+\tau}|)\right] &= \frac{\varphi}{2\pi} n\mu(\mathcal{D}_R|_0^{r_k+\tau}) \\ &= \frac{\varphi}{2\pi} n e^{-\alpha(R-(r_k+\tau))}(1 + o(1)) \\ &= \frac{\varphi}{2\pi} n e^{-2\alpha\log(k)-\alpha(C-\tau)}(1 + o(1)) \\ &= \Theta\left(\varphi n k^{-2\alpha}\right).\end{aligned}$$

Note that $k = O\left((\varphi n/\log(n))^{1/(2\alpha)}\right)$, which is a precondition of this lemma, implies that $\varphi n k^{-2\alpha} = \Omega\left(\log(n)\right)$. Thus, we can apply the Chernoff bound in Corollary 2.3 to conclude that $|V(\mathcal{S}|_0^{r_k+\tau})| = O\left(\varphi n k^{-2\alpha}\right)$ holds with probability $1 - O\left(n^{-c}\right)$ for any constant $c > 0$. □

Using these results, we can now bound the size of the search space in the central part $\mathcal{S}|_{\rho_I(\varphi)}^{\rho_O}$ of our sector $\mathcal{S}$, yielding the following lemma. We note that the lower bound on φ that is a requirement of the following lemma, is weaker than the one we need for Theorem 5.5.

Lemma 5.16. *Let G be a hyperbolic random graph. For every sector $\mathcal{S}$ of angular width $\varphi = \Omega\left(\log(n)^{2\alpha+1}/n\right)$, the degrees of the vertices in $\mathcal{S}|_{\rho_I(\varphi)}^{\rho_O}$ sum to $O\left(\varphi n\right)$ with high probability.*

Proof. First note that, analogous to the argumentation about sectors in the inner part of the disk, we can choose at most $O(n)$ sectors of width 2φ such that any sector of width φ lies completely in one of them. Thus, the probability that there exists a sector where the sum of the vertex degrees in the central part of the disk is too large, is bounded by the probability that it is too large in at least one of these $O(n)$ sectors (of twice the width). In the following, we show for a single sector $\mathcal{S}$ of angular width $\varphi' = 2\varphi$ that the probability for the sum being too large is $O\left(n^{-2}\right)$. The union bound then yields the claim, that the bound holds for every sector of angular width φ.

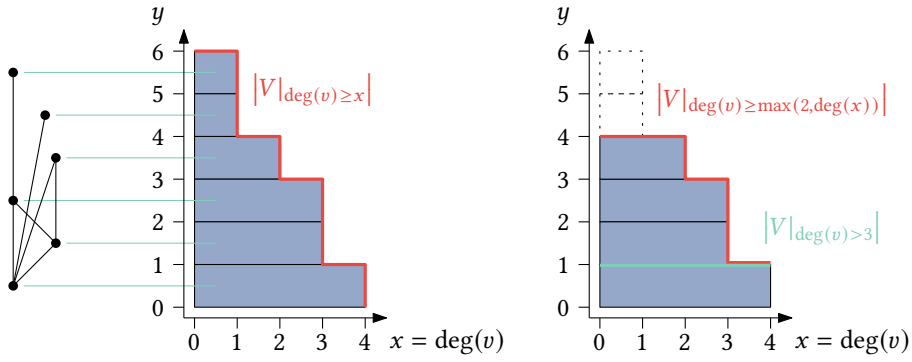


Figure 5.6: (Left) Turning the sum over the degrees into an integral. **(Right)** Summing the degrees of vertices with degree in $[2, 3]$ is equivalent to considering the region up to $x = 3$ and taking the difference of the areas under the red and green curve.

To sum the degrees of all vertices in $\mathcal{S}$, think of a vertex v of degree $\deg(v)$ as a rectangle of height 1 and width $\deg(v)$. For a small graph, Figure 5.6 (left) shows all such rectangles stacked on top of each other, sorted by their degree. Note that the sum of degrees is equal to the area under the function $g(x) = \left|V(\mathcal{S})|_{\deg(v)\geq x}\right|$ where $V(\mathcal{S})|_{\deg(v)\geq x} = \{v \in V(\mathcal{S}) \mid \deg(v) \geq x\}$ is the set of vertices in $\mathcal{S}$ that have degree at least x. Also note that the above considerations do not take into account that we sum only the degrees of vertices in the central part $\mathcal{S}|^{\rho_O}_{\rho_I(\varphi')}$ of $\mathcal{S}$. To resolve this, let $k_{\min}$ and $k_{\max}$ be the minimum and maximum degree of vertices in $\mathcal{S}|^{\rho_O}_{\rho_I(\varphi')}$, respectively. One can see in Figure 5.6 (right) that summing only those degrees that are larger than $k_{\min}$ is equivalent to integrating over $\left|V(\mathcal{S})|_{\deg(v)\geq\max(k_{\min},x)}\right|$ instead of $\left|V(\mathcal{S})|_{\deg(v)\geq x}\right|$. Then, we need to accommodate for the fact that we are only interest in degrees up to $k_{\max}$, which is done by subtracting the area under $\left|V(\mathcal{S})|_{\deg(v)>k_{\max}}\right|$. Thus, we can compute the sum of all degrees as

$$\begin{aligned}\sum_{v\in V\left(\mathcal{S}|^{\rho_O}_{\rho_I(\varphi')}\right)} \deg(v) &\leq \sum_{\substack{v\in V(\mathcal{S}),\\ k_{\min}\leq\deg(v)\leq k_{\max}}} \deg(v)\\ &= \int_0^{k_{\max}} \left|V(\mathcal{S})|_{\deg(v)\geq\max(k_{\min},x)}\right| - \left|V(\mathcal{S})|_{\deg(v)>k_{\max}}\right| \mathrm{d}x\\ &\leq k_{\min}\left|V(\mathcal{S})|_{\deg(v)\geq k_{\min}}\right| + \int_{k_{\min}}^{k_{\max}} \left|V(\mathcal{S})|_{\deg(v)\geq x}\right| \mathrm{d}x.\end{aligned}$$

To compute this integral, we first calculate the minimum and maximum degrees $k_{\min}$ and $k_{\max}$. Afterwards, we apply Lemma 5.15 to bound $\left|V(\mathcal{S})|_{\deg(v)\geq x}\right|$. For $k_{\min}$, assume that vertex v has radius $\rho_O = R - (2+\varepsilon)\log\log(n)$ for any constant $\varepsilon \in (0,1)$. Using Equation (3.9) the expected degree of v is

$$\mathbb{E}\big[\deg(v)\big] = \Theta\Big(ne^{-R/2+(1+\varepsilon/2)\log\log(n)}\Big) = \Theta\Big(\log(n)^{1+\varepsilon/2}\Big).$$

Since $\varepsilon > 0$, this bound is $\omega\big(\log n\big)$, allowing us to apply the Chernoff bounds in Corollary 2.3 and Corollary 2.4 to conclude that $\deg(v) = \Theta\big(\log(n)^{1+\varepsilon/2}\big)$ with high probability. Note that this only holds under the assumption that v has radius exactly ρ_O. However, by Corollary 3.3 vertices with smaller radius have larger expected degree. Thus, $\Theta\big(\log(n)^{1+\varepsilon/2}\big)$ is a lower bound on the expected degrees of all such vertices, allowing us to apply Corollary 2.4 together with a union bound, to conclude that, with high probability, no vertex with smaller radius has smaller degree. Hence, with high probability, the minimum degree in $\mathcal{S}|_{\rho_I(\varphi')}^{\rho_O}$ is

$$k_{\min} = \Theta\Big(\log(n)^{1+\varepsilon/2}\Big).$$

Analogously, the bound on the maximum degree $k_{\max}$ of a vertex in $\mathcal{S}|_{\rho_I(\varphi')}^{\rho_O}$ can be obtained as follows. Let v be a vertex with radius

$$\rho_I(\varphi') = R - 1/\alpha(\log(\varphi'/(2\pi)) + \log(n) - \log\log(n)).$$

By Equation (3.9) the expected degree of v is

$$\mathbb{E}\big[\deg(v)\big] = \Theta\Big((\varphi' n/\log(n))^{1/(2\alpha)}\Big).$$

Since $\varphi' = 2\varphi = \Omega\big(\log(n)^{2\alpha+1}/n\big)$, which is a precondition of this lemma, we can conclude that this bound on the expected degree of v is $\Omega\big(\log(n)\big)$, allowing us to apply Corollary 2.3 to conclude that $\mathbb{E}\big[\deg(v)\big] = \mathrm{O}\big((\varphi n/\log(n))^{1/(2\alpha)}\big)$ holds with high probability. Again, this only holds under the assumption that v has radius exactly $\rho_I(\varphi')$. However, by Corollary 3.3 all vertices with larger radius have smaller expected degree. Therefore, $\mathrm{O}\big((\varphi n/\log(n))^{1/(2\alpha)}\big)$ is a valid upper bound on all their expected degrees, allowing us to apply Corollary 2.3 together with a union bound, to conclude that no vertex with larger radius has larger

degree. Thus, with high probability the maximum degree in $\mathcal{S}|_{\rho_I(\varphi')}^{\rho_O}$ is

$$k_{\max} = \mathrm{O}\left(\left(\frac{\varphi n}{\log(n)}\right)^{1/(2\alpha)}\right).$$

Using Lemma 5.15 we obtain $\left|V(\mathcal{S})|_{\deg(v)\geq x}\right| = \mathrm{O}\big(\varphi n x^{-2\alpha}\big)$ with probability $1-\mathrm{O}(n^{-c})$ for any constant $c > 0$. Note that the requirements $x = \omega\big(\log(n)\big)$ and $x = \mathrm{O}\big((\varphi n/\log(n))^{1/(2\alpha)}\big)$ in Lemma 5.15 are satisfied as $k_{\min} \leq x \leq k_{\max}$. By choosing $c = 2$ and applying the union bound over all degrees, we can conclude that, with high probability

$$\begin{aligned}\sum_{v\in V\left(\mathcal{S}|_{\rho_I(\varphi')}^{\rho_O}\right)} \deg(v) &= \mathrm{O}\Big(\varphi n k_{\min}^{-(2\alpha-1)}\Big) + \mathrm{O}\left(\varphi n \cdot \int_{k_{\min}}^{k_{\max}} x^{-2\alpha}\,\mathrm{d}x\right)\\ &= \mathrm{O}\Big(\varphi n k_{\min}^{-(2\alpha-1)}\Big) + \mathrm{O}\Big(\varphi n \cdot k_{\min}^{-(2\alpha-1)}\big(1-(k_{\min}/k_{\max})^{2\alpha-1}\big)\Big).\end{aligned}$$

As $k_{\min} \leq k_{\max}$ and since $k_{\min} = \omega\big(\log(n)\big)$ this is $\mathrm{O}\big(\varphi n k_{\min}^{-(2\alpha-1)}\big) = \mathrm{O}\big(\varphi n\big)$. □

It remains to bound the sum of the degrees of vertices in the central part of the disk $\mathcal{D}_R|_{\rho_I(2\pi)}^{\rho_O}$ that lie in the neighborhood of a vertex v with radius ρ, i.e., vertices lying in $\mathcal{D}_R|_{\rho_I(2\pi)}^{\rho_O} \cap \mathcal{D}_R(v)$. Similar to the bounds for a sector $\mathcal{S}$, we bound the sum of degrees in $\mathcal{D}_R(v)$ by bounding the number of vertices with a fixed degree k for every possible value of k. If all these bounds hold with probability $1-\mathrm{O}\big(n^{-3}\big)$, then the union bound shows that the sum is concentrated with probability $1-\mathrm{O}\big(n^{-2}\big)$. To obtain a bound that holds for every possible angular coordinate of v (as claimed in Section 5.2.2), we apply Lemma 3.10. There, we choose the random variables X_w to represent the degrees of the vertices. Our bound on the sum that holds with probability $1-\mathrm{O}\big(n^{-2}\big)$ at a fixed angular coordinate, can then be translated to the same asymptotic bound that holds with probability $1-\mathrm{O}\big(n^{-1}\big)$ at every possible angular coordinate.

For a fixed degree $k = \omega\big(\log(n)\big)$, all vertices with degree at least k have radius at most $r_k + \tau$ with high probability due to Lemma 5.14, where $r_k = 2\log(n/k)$ and τ is constant. Thus, all vertices of degree at least k in $\mathcal{D}_R(v)$ lie in $\mathcal{D}_R(v)|_0^{r_k+\tau}$, with high probability. In analogy to Lemma 5.15, we obtain the following bound on the number of vertices in $\mathcal{D}_R(v)|_0^{r_k+\tau}$.

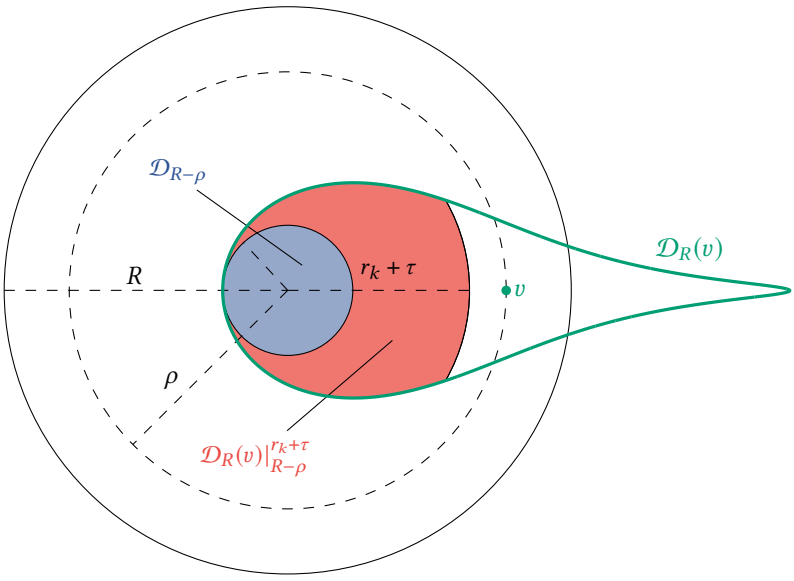


Figure 5.7: Determining the sum of degrees of the neighbors of vertex v that are all contained in $\mathcal{D}_R(v)$ (green). To compute the measure of $\mathcal{D}_R(v)|_0^{r_k+\tau}$ we divide it into two regions $\mathcal{D}_R(v)|_0^{R-\rho} = \mathcal{D}_{R-\rho}$ (blue) and $\mathcal{D}_R(v)|_{R-\rho}^{r_k+\tau}$ (red).

Lemma 5.17. *Let G be a hyperbolic random graph and let v be a vertex with radius $\rho = 1/\alpha(\log(n) - \log\log(n))$. If $k = \omega\big(\log(n)\big)$, the number of neighbors of v with degree at least k is*

$$|\{w \in N(v) \mid \deg(w) \geq k\}| = \mathrm{O}\Big(n^{1-1/(2\alpha)} \log(n)^{1/(2\alpha)} k^{-(2\alpha-1)} + \log(n)\Big)$$

with probability $1 - \mathrm{O}\big(n^{-c}\big)$ for any constant $c > 0$.

Proof. Since $k = \omega\big(\log(n)\big)$, we can apply Lemma 5.14 stating that all vertices of degree at least k in $\mathcal{D}_R(v)$ lie within $\mathcal{D}_R(v)|_0^{r_k+\tau}$ with high probability. To bound the number of neighbors of v with degree at least k we first compute the measure $\mu(\mathcal{D}_R(v)|_0^{r_k+\tau})$. To do this, we separate $\mathcal{D}_R(v)|_0^{r_k+\tau}$ into the disk $\mathcal{D}_R(v)|_0^{R-\rho} = \mathcal{D}_{R-\rho}$ and $\mathcal{D}_R(v)|_{R-\rho}^{r_k+\tau}$; see Figure 5.7. Due to Equation (3.8), we have

$$\mu(\mathcal{D}_{R-\rho}) = \mathrm{O}\big(e^{-\alpha(R-(R-\rho))}\big) = \mathrm{O}\big(\log(n)/n\big),$$

which is already an upper bound on $\mu(\mathcal{D}_R(v)|_0^{r_k+\tau})$ for the case where $r_k + \tau \leq R - \rho$. When $r_k + \tau > R - \rho$, we need to add the measure of $\mathcal{D}_R(v)|_{R-\rho}^{r_k+\tau}$, which is

$$\mu\Big(\mathcal{D}_R(v)|_{R-\rho}^{r_k+\tau}\Big) = \int_{R-\rho}^{r_k+\tau} 2\int_0^{\theta(\rho,r)} f(r,\varphi)\,\mathrm{d}\varphi\,\mathrm{d}r = \mathrm{O}\left(\int_{R-\rho}^{r_k+\tau} \theta(\rho,r) f(r)\,\mathrm{d}r\right)$$

Since $r \in [R-\rho, r_k+\tau]$ in the integral, we have $r \geq R-\rho$, allowing us to apply Equation (3.10) to conclude that $\theta(\rho,r) = \mathrm{O}\big(e^{(R-\rho-r)/2}\big)$. Furthermore, we can bound $f(r) = \mathrm{O}\big(e^{-\alpha(R-r)}\big)$ by Equation (3.7) and obtain

$$\begin{aligned}
\mu\Big(\mathcal{D}_R(v)|_{R-\rho}^{r_k+\tau}\Big) &= \mathrm{O}\left(\int_{R-\rho}^{r_k+\tau} e^{(R-\rho-r)/2}\cdot e^{-\alpha(R-r)}\,\mathrm{d}r\right)\\
&= \mathrm{O}\left(e^{(R-\rho)/2}\cdot e^{-\alpha R}\cdot\int_{R-\rho}^{r_k+\tau} e^{(\alpha-1/2)r}\,\mathrm{d}r\right)\\
&= \mathrm{O}\left(e^{-(\alpha-1/2)R}\cdot e^{-\rho/2}\cdot\left[e^{(\alpha-1/2)(r_k+\tau)} - e^{(\alpha-1/2)(R-\rho)}\right]\right).
\end{aligned}$$

Dropping the negative term in the brackets and substituting $R = 2\log(n) + C$, $\rho = {}^1\!/\!_\alpha(\log(n) - \log\log(n))$, and $r_k = 2\log(n/k)$, we obtain

$$\begin{aligned}
\mu\Big(\mathcal{D}_R(v)|_{R-\rho}^{r_k+\tau}\Big) &= \mathrm{O}\Big(e^{-(\alpha-1/2)R}\cdot e^{-\rho/2}\cdot e^{(\alpha-1/2)(r_k+\tau)}\Big)\\
&= \mathrm{O}\Big(n^{-(2\alpha-1)}\cdot n^{-1/(2\alpha)}\log(n)^{1/(2\alpha)}\cdot n^{2\alpha-1}\cdot k^{-(2\alpha-1)}\Big)\\
&= \mathrm{O}\Big((\log(n)/n)^{1/(2\alpha)}\cdot k^{-(2\alpha-1)}\Big).
\end{aligned}$$

The expected number of vertices in $\mathcal{D}_R(v)|_0^{r_k+\tau}$ is now obtained by reversing the previous split and adding the measures of $\mathcal{D}_{R-\rho}$ and $\mathcal{D}_R(v)|_{R-\rho}^{r_k+\tau}$, which yields

$$\begin{aligned}
\mathbb{E}\big[|V(\mathcal{D}_R(v)|_0^{r_k+\tau})|\big] &= n\cdot\left(\mu\Big(\mathcal{D}_{R-\rho}\Big) + \mu\Big(\mathcal{D}_R(v)|_{R-\rho}^{r_k+\tau}\Big)\right)\\
&= \mathrm{O}\Big(\log(n) + n^{1-1/(2\alpha)}\log(n)^{1/(2\alpha)}k^{-(2\alpha-1)}\Big).
\end{aligned}$$

Clearly, this bound is at least logarithmic. Thus, we can apply Corollary 2.3 to conclude that it holds with probability $1 - \mathrm{O}\big(n^{-c}\big)$ for any constant c. □

With this, we are now ready to bound the sum of the degrees of the vertices in the central part of the disk that are in the neighborhood of a vertex with radius ρ. The proof of the following lemma is analogous to the one of Lemma 5.16.

Lemma 5.18. *Let G be a hyperbolic random graph and let v be a hypothetical vertex with radius $\rho = 1/\alpha(\log(n) - \log\log(n))$ and arbitrary angular coordinate. The degrees of neighbors of v in $\mathcal{D}_R|_{\rho_I(2\pi)}^{\rho_O}$ sum to $\tilde{\mathrm{O}}\big(n^{1/(2\alpha)}\big)$ with high probability.*

Proof. Recall that $\mathcal{D}_R(v)$ is the disk containing all neighbors of v. To bound the sum of the degrees of the vertices in $\mathcal{D}_R(v)|_{\rho_I(2\pi)}^{\rho_O}$, we use basically the same proof as in Lemma 5.16 except we use Lemma 5.17 instead of Lemma 5.15. Thus,

$$\sum_{w \in V\left(\mathcal{D}_R(v)|_{\rho_I(2\pi)}^{\rho_O}\right)} \deg(w) \leq k_{\min}\left|V(\mathcal{D}_R(v))|_{\deg(v)\geq k_{\min}}\right| + \int_{k_{\min}}^{k_{\max}} \left|V(\mathcal{D}_R(v))|_{\deg(v)\geq x}\right| \mathrm{d}x,$$

where $V(\mathcal{D}_R(v))|_{\deg(v)\geq x}$ is the set of vertices of degree at least x in $\mathcal{D}_R(v)$ and $k_{\min}$ and $k_{\max}$ are the minimum and maximum degree in $\mathcal{D}_R(v)|_{\rho_I(2\pi)}^{\rho_O}$, respectively.

We start with computing $k_{\min}$ and $k_{\max}$. Using Equation (3.9), Corollary 2.3, and Corollary 2.4, we obtain that a vertex of radius $\rho_O = R - (2+\varepsilon)\log\log(n)$, for any $\varepsilon \in (0,1)$, has degree $k_{\min} = \Theta\big(\log(n)^{1+\varepsilon/2}\big)$ with high probability. Moreover, by the same argumentation as in the proof of Lemma 5.16 no vertex with smaller radius has smaller degree, with high probability. Additionally, a vertex with radius $\rho_I(2\pi) = R - 1/\alpha(\log(n) - \log\log(n))$ has degree $k_{\max} = \mathrm{O}\Big((n/\log(n))^{1/(2\alpha)}\Big)$ and no vertex with larger radius has larger degree, with high probability. It follows that we can use the bound on $\left|V(\mathcal{D}_R(v))|_{\deg(v)\geq x}\right|$ established in Lemma 5.17. Thus, we obtain

$$\sum_{w \in V\left(\mathcal{D}_R(v)|_{\rho_I(2\pi)}^{\rho_O}\right)} \deg(w) = \tilde{\mathrm{O}}\Big(k_{\min} \cdot n^{1-1/(2\alpha)} k_{\min}^{-(2\alpha-1)}\Big) + \tilde{\mathrm{O}}\left(n^{1-1/(2\alpha)} \int_{k_{\min}}^{k_{\max}} x^{-(2\alpha-1)}\,\mathrm{d}x\right).$$

Replacing $k_{\min}$ and simplifying the first term in the sum yields $\tilde{\mathrm{O}}\big(n^{1-1/(2\alpha)}\big)$,

which is smaller than the claimed bound. For the second term, we obtain

$$\tilde{\mathrm{O}}\left(n^{1-1/(2\alpha)} \int_{k_{\min}}^{k_{\max}} x^{-(2\alpha-1)}\,\mathrm{d}x\right) = \tilde{\mathrm{O}}\left(n^{1-1/(2\alpha)}\left[k_{\max}^{2-2\alpha} - k_{\min}^{2-2\alpha}\right]\right).$$

Finally, dropping the negative term and replacing $k_{\max} = \tilde{\mathrm{O}}\left(n^{1/(2\alpha)}\right)$, we obtain $\tilde{\mathrm{O}}\left(n^{1-1/(2\alpha)+1/\alpha-1}\right) = \tilde{\mathrm{O}}\left(n^{1/(2\alpha)}\right)$. □

5.3.3 The Outer Part of the Disk

At this point we have bounded the sum of the degrees of the vertices with radius at most $\rho_O = R - (2+\varepsilon)\log\log(n)$ (for any constant $\varepsilon \in (0,1)$) that lie in a sector of angular width $\varphi = \Omega\left(\log(n)^{2\alpha+1}/n\right)$ or in the neighborhood of a vertex with radius ρ. It remains to bound the sums when considering vertices with radii larger than ρ_O. We start by computing the expected value of the sum of vertex degrees in the outer part of a sector $\mathcal{S}|_{\rho_O}^{R}$.

Lemma 5.19. *Let G be a hyperbolic random graph. For a sector $\mathcal{S}$ of angular width φ, the degrees of vertices in $\mathcal{S}|_{\rho_O}^{R}$ sum to $\Theta\left(\varphi n\right)$ in expectation.*

Proof. Let $\deg(v)$ be the random variable describing the degree of a vertex v. Moreover, let X_v be the indicator variable that is 1 if $v \in V(\mathcal{S}|_{\rho_O}^{R})$ and 0 otherwise. Then the expected sum of the degrees of vertices in $\mathcal{S}|_{\rho_O}^{R}$ is given by

$$\begin{aligned}\mathbb{E}\left[\sum_{v\in V} X_v \cdot \deg(v)\right] &= \sum_{v\in V} \mathbb{E}\left[X_v \cdot \deg(v)\right] \\ &= n \cdot \Pr\left[v \in V\left(\mathcal{S}|_{\rho_O}^{R}\right)\right] \cdot \mathbb{E}\left[\deg(v) \,\middle|\, v \in V\left(\mathcal{S}|_{\rho_O}^{R}\right)\right].\end{aligned}$$

Note that $\Pr\left[v \in V(\mathcal{S}|_{\rho_O}^{R})\right]$ is equivalent to the measure $\mu(\mathcal{S}|_{\rho_O}^{R})$. As the angular coordinates are distributed uniformly at random, the whole sector $\mathcal{S}$ has a measure of $\Theta\left(\varphi\right)$. Moreover, the region of the disk containing the points with constant distance to the boundary has constant measure. As a consequence, the measure of $\mathcal{S}|_{\rho_O}^{R}$ is also $\Theta\left(\varphi\right)$. For the sake of completeness, we formally compute the measure of $\mathcal{S}|_{\rho_O}^{R}$ by considering a complete sector and subtracting

the inner part, which yields

$$\begin{aligned}
\mu\left(\mathcal{S}|_{\rho_O}^{R}\right) &= \mu\left(\mathcal{S} \setminus \mathcal{S}|_{0}^{\rho_O}\right) \\
&= \frac{\varphi}{2\pi}\left(1 - \mu\left(\mathcal{D}_{\rho_O}\right)\right) \\
&= \frac{\varphi}{2\pi}\left(1 - e^{-\alpha(R-\rho_O)}\left(1 + \mathrm{o}(1)\right)\right) \\
&= \frac{\varphi}{2\pi}\left(1 - \mathrm{O}\left(\log(n)^{-\alpha(2+\varepsilon)}\right)\right) \\
&= \Theta(\varphi).
\end{aligned}$$

It remains to determine $\mathbb{E}\left[\deg(v) \,\middle|\, v \in V(\mathcal{S}|_{\rho_O}^{R})\right]$, which can be done as follows.

$$\begin{aligned}
\mathbb{E}\left[\deg(v) \,\middle|\, v \in V\left(\mathcal{S}|_{\rho_O}^{R}\right)\right] &= \iint_{\mathcal{S}|_{\rho_O}^{R}} \mathbb{E}\left[\deg(v) \,\middle|\, r(v) = r\right] \frac{f(r,\phi)}{\mu\left(\mathcal{S}|_{\rho_O}^{R}\right)} \,\mathrm{d}\phi \,\mathrm{d}r \\
&= \frac{1}{\mu\left(\mathcal{S}|_{\rho_O}^{R}\right)} \int_{\rho_O}^{R} \int_{0}^{\varphi} \mathbb{E}\left[\deg(v) \,\middle|\, r(v) = r\right] f(r,\phi) \,\mathrm{d}\phi \,\mathrm{d}r \\
&= \Theta(1) \cdot \int_{\rho_O}^{R} \mathbb{E}\left[\deg(v) \,\middle|\, r(v) = r\right] f(r) \,\mathrm{d}r \\
&= \Theta(1) \cdot n \cdot e^{-\alpha R} \int_{\rho_O}^{R} e^{(\alpha - 1/2)r} \,\mathrm{d}r \\
&= \Theta(1) \cdot n \cdot e^{-\alpha R}\left[e^{(\alpha-1/2)R} - e^{(\alpha-1/2)\rho_O}\right] \\
&= \Theta(1) \cdot n \cdot e^{-R/2}\left[1 - e^{-(\alpha-1/2)(R-\rho_O)}\right]
\end{aligned}$$

Note that the part in brackets can be bounded by a constant. Moreover, since $R = 2\log(n) + C$, we know that $n \cdot e^{-R/2}$ is constant as well. It follows that $\mathbb{E}\left[\deg(v) \,\middle|\, v \in V(\mathcal{S}|_{\rho_O}^{R})\right]$ is in $\Theta(1)$. Consequently, the expected sum of the degrees is $\Theta(\varphi n)$. □

Unfortunately, the sum of the vertex degrees in $\mathcal{S}|_{\rho_O}^{R}$ is not concentrated sufficiently well around its expectation to conclude that this bound also holds with high probability. The problem lies with the high-degree vertices in the graph, which can be adjacent to none or all vertices in $\mathcal{S}|_{\rho_O}^{R}$ depending on their

positions. That is, small perturbations of the position of a single high-degree vertex can change the sum by too much. To overcome this issue, we consider the impact of high-degree vertices separately. To this end, we partition the edge set that contributes to the degrees of the vertices in $\mathcal{S}|_{\rho_O}^R$ into two sets E_I and E_O, denoting the *inner edges* where the other endpoint is in $\mathcal{D}_R|_0^{\rho_O}$ and the *outer edges* where the other endpoint is in $\mathcal{D}_R|_{\rho_O}^R$, respectively. The sum of the degrees of the vertices in $\mathcal{S}|_{\rho_O}^R$ can then be bounded by taking the number of inner edges and adding them to twice the number of outer edges. That is,

$$\sum_{v \in V\left(\mathcal{S}|_{\rho_O}^R\right)} \deg(v) \leq |E_I| + 2|E_O|.$$

Since E_I denotes all edges with one endpoint in $\mathcal{S}|_{\rho_O}^R$ and the other in the inner or central part of the disk, we can obtain an upper bound on the first summand by summing the degrees of the vertices in $\mathcal{D}_R|_0^{\rho_O}$ that are adjacent to any vertex in $\mathcal{S}|_{\rho_O}^R$. Since $\rho \leq \rho_O$, we have $\mathcal{S}|_{\rho_O}^R \subseteq \mathcal{S}|_{\rho}^R$, allowing us to apply Lemma 5.7 to conclude that all such vertices are contained in $\mathcal{S}$ or are neighbors of the two hypothetical corner vertices c_1 and c_2, which both have radius ρ. Thus, $|E_I|$ can be bounded by the sum of the degrees of vertices in a sector and in the neighborhood of a vertex with radius ρ, but constrained to vertices in the inner and central parts of the disk. Corresponding bounds that hold with high probability have been determined above. For the sector we obtain an upper bound of $\tilde{\mathrm{O}}\big(\deg_{\max}(G)\big)$ for the inner part (Corollary 5.12) and $\mathrm{O}\big(\varphi n\big)$ for the central part (Lemma 5.16). For the neighborhood of a vertex with radius ρ we have $\tilde{\mathrm{O}}\big(\deg_{\max}(G)\big)$ for the inner part (Corollary 5.13) and $\tilde{\mathrm{O}}\big(n^{1/(2\alpha)}\big)$ for the central part (Lemma 5.18). Taking them together, we obtain the following.

Corollary 5.20. *Let G be a hyperbolic random graph. For every sector $\mathcal{S}$ of angular width $\varphi = \Omega\big(\log(n)^{2\alpha+1}/n\big)$, the number of edges with one endpoint in $\mathcal{S}|_{\rho_O}^R$ and the other in $\mathcal{D}_R|_0^{\rho_O}$ is $\tilde{\mathrm{O}}\big(\varphi n + n^{1/(2\alpha)} + \deg_{\max}(G)\big)$, with high probability.*

To obtain an upper bound on the second part of the above sum, we aim to apply a method of typical bounded differences (see Section 2.2.3) based on the fact that changing the position of a single vertex has typically only little impact on the number of outer edges. The idea is as follows. We consider $|E_O|$ as a function that only depends on the random variables $P_1, \dots, P_n$ denoting the positions of the vertices in the graph and we ask ourselves: How much can $|E_O|$

change, if we alter the position of a single vertex i? Clearly, this change can be large in the worst case. Assume that we move i from outside $\mathcal{D}_R|^R_{\rho_O}$ into $\mathcal{S}|^R_{\rho_O}$. Then, i does not contribute anything to $|E_O|$ before the move and the increase in $|E_O|$ depends on the number of outer edges that are incident to i after the move, which can be $n-1$ in the worst case. However, it is very unlikely that a vertex in $\mathcal{S}|^R_{\rho_O}$ has this many neighbors that lie in the outer part of the disk. In fact, its degree is typically much smaller. To formalize this, we represent the typical case using an event A, denoting that the degree of such a vertex is at most a constant factor larger than the expected degree of a vertex with radius $\rho_O = R - (2+\varepsilon)\log\log(n)$ for any constant $\varepsilon \in (0,1)$. More precisely, A denotes the event in which all disks of radius R with center in $\mathcal{D}_R|^R_{\rho_O}$ contain at most $\mathrm{O}\big(\log(n)^{1+\varepsilon/2}\big)$ vertices. In this case, moving a vertex i in the same way as before leads to a much smaller increase in the number of outer edges. Assuming that A holds before the move, there are at most $\mathrm{O}\big(\log(n)^{1+\varepsilon/2}\big)$ outer edges incident to i after the move, which corresponds to the increase of $|E_O|$. The following lemma defines the event A formally and shows that it holds with high probability.

Lemma 5.21. *Let G be a hyperbolic random graph and let $\rho_O = R - (2+\varepsilon)\cdot\log\log(n)$ for any constant $\varepsilon \in (0,1)$. Then, all disks $\mathcal{D}$ with radius R and center in $\mathcal{D}_R|^R_{\rho_O}$ contain at most $|V(\mathcal{D})| = \mathrm{O}\big(\log(n)^{1+\varepsilon/2}\big)$ vertices, with probability $1-\mathrm{O}\big(n^{-c}\big)$ for any constant c.*

Proof. Let $\mathcal{D}$ be a disk of radius R and center $\mathcal{P} \in \mathcal{D}_R|^R_{\rho_O}$. By Corollary 3.3, a valid upper bound on the expected number of vertices in $\mathcal{D}$ can be obtained by considering the disk $\mathcal{D}'$ at center $\mathcal{P}'$ instead, which has the same angular coordinate as $\mathcal{P}$ and radius ρ_O. The expected number of vertices in $\mathcal{D}'$ can be computed using Equation (3.9), which yields

$$\mathbb{E}\big[|V(\mathcal{D})|\big] \leq \mathbb{E}\big[|V(\mathcal{D}')|\big] = \mathrm{O}\Big(ne^{-\rho_O/2}\Big) = \mathrm{O}\Big(\log(n)^{1+\varepsilon/2}\Big).$$

Moreover, since $\varepsilon > 0$, this bound is $\omega\big(\log(n)\big)$ and we can apply Corollary 2.3 to conclude that $|V(\mathcal{D})| = \mathrm{O}\big(\log(n)^{1+\varepsilon/2}\big)$ holds with probability $1-\mathrm{O}\big(n^{-c'}\big)$ for any constant c'. To obtain a bound that holds for every possible angular coordinate for $\mathcal{P}$, we apply Lemma 3.10, which allows us to translate our bound that holds for any given disk $\mathcal{D}$ with probability $1-\mathrm{O}\big(n^{-c'}\big)$ to the same asymptotic bound that holds with probability $1-\mathrm{O}\big(n^{-c'+1}\big)$ for all possible angular coordinates. Choosing $c' = c+1$ then yields the claim. □

So while moving a single vertex leads to a large change in the number of outer edges $|E_O|$ in the worst case, we observe only small changes in the typical case A. We are now ready to bound the number $|E_O|$ of outer edges, i.e, edges that are incident to vertices in a sector $\mathcal{S}|_{\rho_O}^R$ and have their other endpoint in $\mathcal{D}_R|_{\rho_O}^R$.

Lemma 5.22. *Let G be a hyperbolic random graph. For every sector $\mathcal{S}$ of angular width $\varphi = \Omega\big(\log(n)^2/n^{1/2}\big)$, the number of edges with one endpoint in $\mathcal{S}|_{\rho_O}^R$ and the other in $\mathcal{D}_R|_{\rho_O}^R$ is $\mathrm{O}\big(\varphi n\big)$, with high probability.*

Proof. First note that, analogous to the proof of Lemma 5.16, we can cover the disk with $\mathrm{O}(n)$ sectors of angular width 2φ such that any sector of angular width φ lies completely in one of them. In the following, we show that the claimed bound holds with probability $\mathrm{O}\big(n^{-2}\big)$ for a single sector $\mathcal{S}$ of twice the width[5]. Applying the union bound then yields the claim.

We consider $|E_O|$, the number of edges with one endpoint in $\mathcal{S}|_{\rho_O}^R$ and the other in $\mathcal{D}_R|_{\rho_O}^R$, as a function hat only depends on the random variables $P_1, \dots, P_n$ denoting the positions of the vertices in the graph. To show that $|E_O|$ does not exceed an upper bound with high probability, we aim to apply the method of typical bounded differences (Corollary 2.8). We represent the typical case with an event A, denoting that all disks $\mathcal{D}$ of radius R and center in $\mathcal{D}_R|_{\rho_O}^R$ contain at most $\mathrm{O}\big(\log(n)^{1+\varepsilon/2}\big)$ vertices for any constant $\varepsilon \in (0,1)$. In order to determine the parameters $\Delta_i^A \leq \Delta_i$ for $i \in [n]$ with which $|E_O|$ fulfills the typical bounded differences condition with respect to A (Equation (2.3)), we have to bound the maximum change in $|E_O|$ obtained by moving a single vertex. As argued before, this change is at most $\Delta_i = n - 1$ for all $i \in [n]$ in the worst case. To bound the Δ_i^A, we start with a configuration of vertex coordinates in which the event A holds. In this case, it is easy to see that moving a single vertex i changes $|E_O|$ by at most $\Delta_i^A = \mathrm{O}\big(\log(n)^{1+\varepsilon/2}\big)$ for all $i \in [n]$, since the degree of i is at most this large after the move and so is the number of outer edges it contributes to $|E_O|$.

We are now ready to apply the method of typical bounded differences (Corollary 2.8). For an upper bound $g(n)$ on $|E_O|$, any constant $c > 1$, and all $\varepsilon_1, \dots, \varepsilon_n \in (0,1]$ it states that

$$\Pr\big[|E_O| > cg(n)\big] \leq e^{-((c-1)g(n))^2/(2\Delta)} + \Pr[\neg A] \sum_{i\in[n]} 1/\varepsilon_i,$$

5 We note that this factor 2 vanishes in the asymptotics throughout the proof.

where $\Delta = \sum_{i\in[n]}(\Delta_i^A + \varepsilon_i(\Delta_i - \Delta_i^A))^2$. First note that a valid upper bound on the expected number of outer edges incident to vertices in $\mathcal{S}|_{\rho_O}^R$ is given by the expected sum of the degrees of these vertices. Thus, by Lemma 5.19 we can choose $g(n) = \Theta(\varphi n)$. Moreover, by choosing $\varepsilon_i = 1/n$ for all $i \in [n]$ and since $\Delta_i = n - 1$ and $\Delta_i^A = O(\log(n)^{1+\varepsilon/2})$ for all $i \in [n]$, we can compute Δ as

$$\begin{aligned}
\Delta &= \sum_{i\in[n]} (\Delta_i^A + \varepsilon_i(\Delta_i - \Delta_i^A))^2 \\
&= O\left(n \cdot \left(\log(n)^{1+\varepsilon/2} + 1/n(n - \log(n)^{1+\varepsilon/2})\right)^2\right) \\
&= O\left(n \cdot \left(\log(n)^{1+\varepsilon/2} + (1 - o(1))\right)^2\right) \\
&= O\left(n \cdot \log(n)^{2+\varepsilon}\right)
\end{aligned}$$

Consequently, the above probability can be bounded by

$$\begin{aligned}
\Pr\left[|E_O| > cg(n)\right] &\leq \exp\left(-\Theta\left(\frac{(\varphi n)^2}{n\log(n)^{2+\varepsilon}}\right)\right) + \Pr[\neg A] \cdot n^2 \\
&\leq \exp\left(-\Theta\left(\frac{\varphi^2 n}{\log(n)^{2+\varepsilon}}\right)\right) + \Pr[\neg A] \cdot n^2.
\end{aligned}$$

Since $\varphi = \Omega(\log(n)^2/n^{1/2})$ is a precondition of this lemma and since $\varepsilon < 1$, we can conclude that the fraction is $\omega(\log(n))$, which means that the first summand is $O(n^{-c'})$ for any constant c'. Moreover, by Lemma 5.21 event A holds with probability $1 - O(n^{-c'})$ for any constant c'. Choosing $c' = 3$ then yields the claim. □

5.3.4 The Complete Disk

Having obtained the required bounds for the inner, central, and outer parts of the disk, we can now combine them to bound the sum of the degrees in a sector and in the neighborhoods of the hypothetical corner vertices. We start with Theorem 5.5, which bounds the sum of degrees in a sector. To improve readability, we restate the theorem here.

Theorem 5.5. *Let G be a hyperbolic random graph with n vertices. Then, the degrees of vertices in every sector of angular width $\varphi = \Omega\big(\log(n)^2/n^{1/2}\big)$ sum to $\tilde{O}\big(\varphi n + n^{1/(2\alpha)} + \deg_{\max}(G)\big)$ with high probability.*

Proof. For the inner and central parts of every sector the sum of the vertex degrees is bounded by $\tilde{O}\big(\varphi n + \deg_{\max}(G)\big)$ with high probability due to Corollary 5.12 and Lemma 5.16. As argued above, the sum of the degrees of the remaining vertices, i.e., vertices with radius at least ρ_O, can be bounded by counting the number of inner edges and adding twice the number of outer edges. Since $\varphi = \Omega\big(\log(n)^2/n^{1/2}\big)$, we can apply Corollary 5.20 and Lemma 5.22 to conclude that the corresponding sum is bounded by $\tilde{O}\big(\varphi n + n^{1/(2\alpha)} + \deg_{\max}(G)\big)$, with high probability. □

Lastly, it remains to bound the sum of the degrees of the neighbors of the hypothetical corner vertices that were used to bound the size of the search space in the second phase. Again, for the sake of readability, we restate the corresponding lemma here.

Lemma 5.9. *Let G be a hyperbolic random graph and let v be a hypothetical vertex with radius $\rho = 1/\alpha(\log n - \log\log n)$ and arbitrary angular coordinate. Then, the degrees of the neighbors of v sum to $\tilde{O}\big(n^{2-1/\alpha} + n^{1/(2\alpha)} + \deg_{\max}(G)\big)$ with high probability.*

Proof. For the inner and central parts of the neighborhood of a vertex with radius ρ and arbitrary angular coordinate the sum of the degrees is bounded by $\tilde{O}\big(\deg_{\max}(G) + n^{1/(2\alpha)}\big)$ with high probability, as a result of Corollary 5.13 and Lemma 5.18. For the sum of the degrees in the outer part of the disk, note that all neighbors of radius at least ρ have angular distance at most $\varphi = O\big(n^{-(1/\alpha-1)}\big)$, as explained in Section 5.2.2. As a consequence, we can use Theorem 5.5 to conclude that the claimed bound holds for the sum of their degrees. Note that if φ is too small to meet the requirements of Theorem 5.5, we can choose $\varphi = \tilde{O}\big(n^{-1/2}\big)$ as a valid upper bound instead, in order to conclude that the sum of degrees in the outer part of the neighborhood is $\tilde{O}\big(n^{1/2}\big)$, which is $\tilde{O}\big(n^{1/(2\alpha)}\big)$ for all $\alpha \in (1/2, 1)$. □

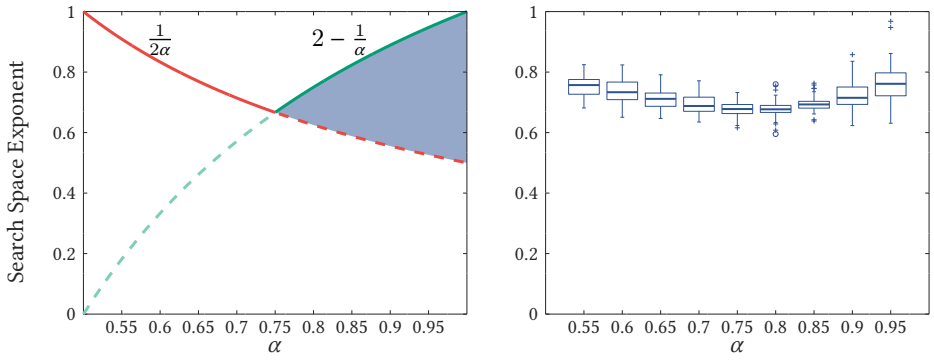


Figure 5.8: (Left) The exponent of our theoretical bound depending on α. **(Right)** The corresponding empirically measured search spaces. The data was obtained by generating 20 hyperbolic random graphs with average degree roughly 8 for each shown α and each $n \in \{100\text{k}, 200\text{k}, 300\text{k}\}$. For each graph we sampled 300k start–destination pairs and report the maximum number of edges explored in one search. The numbers are normalized with the total number of edges m of the graph such that x is plotted for a search space of size m^x.

5.4 Discussion

In the following, we briefly discuss why we think that the running time bound of $\tilde{\mathrm{O}}\big(n^{2-1/\alpha} + n^{1/(2\alpha)} + \deg_{\max}(G)\big)$ is rather tight; see Figure 5.8 (left) for a plot of the exponents. Clearly, the maximum degree of the graph is a lower bound, i.e., we cannot improve the $\deg_{\max}(G)$. As $\deg_{\max}(G) = \tilde{\Theta}\big(n^{1/(2\alpha)}\big)$ holds almost surely [GPP12], we also cannot improve below $\tilde{\mathrm{O}}\big(n^{1/(2\alpha)}\big)$. For the term $n^{2-1/\alpha}$ we do not have a lower bound. Thus, the blue region in Figure 5.8 (left) is the only part where our bound can potentially be improved. However, by only making a single step from a vertex with radius $\rho = 1/\alpha(\log(n) - \log\log(n))$, we can already reach vertices with angular distance $\Theta\big(n^{-(1/\alpha-1)}\big)$. Thus, it seems likely, that there exists a start–destination pair such that all vertices within a sector of this angular width are actually explored. As such a sector contains $\Theta\big(n^{2-1/\alpha}\big)$ vertices, our bound seems rather tight (at least asymptotically and up to poly-logarithmic factors). For a comparison of our theoretical bound with actual search-space sizes in hyperbolic random graphs; see Figure 5.8.

Finally, in order to put our results into perspective, we discuss the following question: How does a heterogeneous degree distribution impact the exponent in

the running time of the bidirectional BFS? First, considering networks with no underlying geometry, the exponent is $1/2$ for homogeneous networks and $(4-\beta)/2 = 3/2 - \alpha$ for heterogeneous networks with power-law exponent β [BN16]. That is, when increasing the heterogeneity by letting α go from 1 to $1/2$, the exponent increases from $1/2$ to 1. This can be explained by the fact that a heterogeneous degree distribution leads to high-degree vertices, which leads to a higher running time when they are explored.

On hyperbolic random graphs, we get the same effect. The $1/(2\alpha)$-part of the exponent (the red function in Figure 5.8) is very similar to the above $3/2 - \alpha$. However, due to the underlying geometry, the heterogeneity has another effect, expressed by the $2 - 1/\alpha$-part of the exponent (the green function in Figure 5.8). This can be explained as follows. The underlying geometry constrains the parts of the graph that a vertex can connect to. As a result, the search space cannot expand sufficiently fast on homogeneous networks and we only get a constant speed-up, i.e., the exponent is 1. However, increasing the heterogeneity leads to high degree vertices, which accelerate the expansion of the search spaces, leading to a lower exponent.

In conclusion, we can say that heterogeneity has two effects on the bidirectional BFS:

1. More heterogeneity leads to higher running times as exploring high degree-vertices is costly.

2. More heterogeneity leads to lower running times as high degree-vertices let the search spaces expand quickly.

For networks without underlying geometry, the second effect is irrelevant, as the search space always expands quickly due to the independence of edges. Thus, the running time is better the more homogeneous the network. For networks with underlying geometry, both effects play an important role leading to the v-shape in Figure 5.8. For high heterogeneity ($\alpha < 0.75$), the cost of exploring high-degree vertices dominates, leading to the exponent $1/(2\alpha)$. For lower heterogeneity ($\alpha > 0.75$), the slower expanding search space due to the underlying geometry dominates, leading to the exponent $2 - 1/\alpha$.

6 Exact Vertex Cover in Hyperbolic Random Graphs

This chapter is based on joint work with Thomas Bläsius, Philipp Fischbeck, and Tobias Friedrich [Blä+21a].

6.1 Introduction

A *vertex cover* of a graph G is a subset S of the vertices that leaves the graph edgeless upon deletion. That is, each edge in G is incident to at least one vertex in S. The problem of finding a *smallest* vertex cover of a graph is highly relevant due to its applications in computational biology [Abu+04], scheduling [ELW16], and internet security [Fil+07]. Unfortunately, since finding a minimum vertex cover is one of the most fundamental NP-complete graph problems [Kar72], there are probably no algorithms that solve it efficiently.

The best known algorithm to compute a minimum vertex cover runs in time $1.1996^n\text{poly}(n)$ [XN17]. We note that this is actually an algorithm to solve the *maximum independent set* problem. There, the goal is to find the largest set of vertices such that no two of them are adjacent. However, since by definition all vertices, that are not in the minimum vertex cover, form an independent set, one can easily derive a maximum independent set from a minimum vertex cover, and vice versa.

To analyze the complexity of vertex cover on a finer scale, several parameterized solutions have been proposed. One can determine whether a graph has a vertex cover of size k by applying a *branch-and-reduce* algorithm. The idea is to build a search tree by recursively considering two possible extensions of the current vertex cover (*branching*), until a vertex cover is found or the size of the current cover exceeds k. Each branching step is followed by a *reduce* step in which *reduction rules* are applied to make the considered graph smaller. This technique yields a simple $O\big(2^k\text{poly}(n)\big)$ algorithm, where the exponential portion comes from the branching. The best known FPT (fixed-parameter tractable) algorithm runs in $O\big(1.2738^k + kn\big)$ time [CKX10], and unless ETH (exponential time hypothesis) fails, there can be no $2^{o(k)}\text{poly}(n)$ algorithm [CJ03].

While these FPT approaches promise relatively small running times if the considered network has a small vertex cover, a recent study shows that the optimal solution is large for many real-world networks [AI16]. However, it was also observed there that applying a branch-and-reduce technique on real instances is very efficient. Some of the considered networks had millions of vertices, yet an optimal solution (also containing millions of vertices) was computed within seconds. Most instances were solved so quickly since the expensive branching was not necessary at all. In fact, the application of the reduction rules alone already yielded an optimal solution. In this chapter, we investigate the performance of one of these rules called the *dominance reduction rule*, which eliminates vertices whose neighborhood contains a vertex together with its neighborhood. It reduces the graph to a very small remainder on which the branching, if necessary, can be done quickly. We trace the effectiveness of the dominance rule back to two properties that are often observed in real-world networks: a *heterogeneous degree distribution* and *high clustering*.

In particular, we use the hyperbolic random graph model (see Section 3.3) to formalize these properties and analyze the performance of the dominance rule on such networks. Our results show that this rule performs sufficiently well such that vertex cover can be solved in polynomial time on hyperbolic random graphs, with high probability. The proof is based on the fact that even a single application of the dominance reduction rule reduces a hyperbolic random graph to a remainder with small pathwidth on which a vertex cover can then be obtained efficiently. We note that, while we focus on hyperbolic random graphs with power-law exponent $\beta = 2\alpha + 1$ for $\alpha \in (1/2, 1)$, it was previously shown for $\beta > 3$, where hyperbolic random graphs degenerate into small components, none having a size linear in n, that the obtained graphs have logarithmic treewidth [BFK16], meaning the vertex cover problem can be solved efficiently in that case.

Our analysis provides an explanation for why the vertex cover problem can be solved efficiently on practical instances. We note that, while our analysis makes use of the underlying hyperbolic geometry, the algorithm itself is oblivious to it. Since our proof relies on certain structural properties of hyperbolic random graphs, we further conducted experiments to test whether these are also found in real-world networks. Our results indicate that these predictions actually match the real world for a significant fraction of networks.

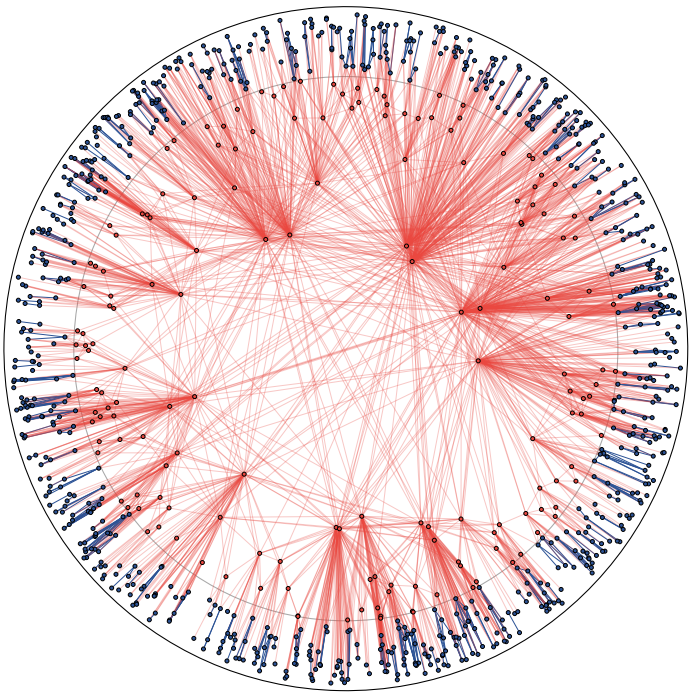


Figure 6.1: A hyperbolic random graph with 979 vertices, average degree 8.3, and a power-law exponent of 2.5. In such a graph the red vertices and edges are removed by the dominance reduction rule, with high probability. Additionally, the remaining subgraph in the outer band (consisting of the blue vertices and edges) has a small path width, with high probability.

6.2 Vertex Cover on Hyperbolic Random Graphs

Reduction rules are often applied as a preprocessing step, before using a brute force search or branching in a search tree. They simplify the input by removing parts that are easy to solve. For example, an isolated vertex does not cover any edges and can thus never be part of a minimum vertex cover. Consequently, in a preprocessing step all isolated vertices can be removed, which leads to a reduced input size without impeding the search for a minimum.

The dominance reduction rule, defined by Garfinkel and Nemhauser [GN72], has been applied to solve the independent set problem [FGK09], and later to compute vertex cover solutions in the experiments by Akiba and Iwata [AI16].

Formally, a vertex u *dominates* a neighbor $v \in N(u)$ if $(N(v) \setminus \{u\}) \subseteq N(u)$, i.e., if all neighbors of v are also neighbors of u. We say that u is *dominant* if it dominates at least one vertex. The dominance rule states that u can be added to the vertex cover (and afterwards be removed from the graph), without impeding the search for a minimum vertex cover. To see that this is correct, assume that u dominates v and let S be a minimum vertex cover that does not contain u. Since S has to cover all edges, it contains all neighbors of u. These neighbors include v and all of v's neighbors, since u dominates v. Therefore, removing v from S leaves only the edge $\{u, v\}$ uncovered, which can be fixed by adding u instead. The resulting vertex cover has the same size as S. When searching for a minimum vertex cover of a graph G, it is thus safe to assume that u is part of the solution and to reduce the search to the induced subgraph $G[V \setminus \{u\}]$, obtained by removing u and its incident edges.

In the remainder of this section, we study the effectiveness of the dominance reduction rule on hyperbolic random graphs and conclude that the vertex cover problem can be solved efficiently on these graphs. Our results are summarized in the following main theorem.

Theorem 6.1. *Let G be a hyperbolic random graph on n vertices. Then the vertex cover problem on G can be solved in* $\text{poly}(n)$ *time, with high probability.*

The proof of Theorem 6.1 consists of two parts that make use of the underlying hyperbolic geometry. In the first part, we show that applying the dominance reduction rule once removes all vertices in the inner part of the hyperbolic disk with high probability, as depicted in Figure 6.1. We note that this is independent of the order in which the reduction rule is applied, as dominant vertices remain dominant after removing other dominant vertices. In the second part, we consider the induced subgraph containing the remaining vertices near the boundary of the disk (blue vertices in Figure 6.1), and show that this part has simple structure, which can be utilized to compute an optimal vertex cover efficiently.

6.2.1 Dominance

Recall that a hyperbolic random graph is obtained by distributing n vertices in a hyperbolic disk $\mathcal{D}_R$ and that any two are adjacent if their distance is at most R (see Section 3.3). Consequently, one can imagine the neighborhood of a vertex u as another disk $\mathcal{D}_R(u)$. Vertex u dominates another vertex v if its neighborhood disk completely contains that of v (assuming the one of v is constrained to $\mathcal{D}_R$),

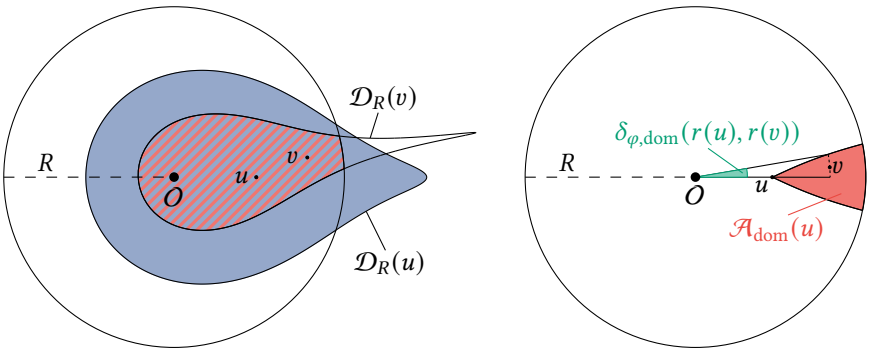


Figure 6.2: (Left) Vertex u dominates vertex v, since $\mathcal{D}_R(v) \cap \mathcal{D}_R$ (shaded red) is completely contained in $\mathcal{D}_R(u)$ (blue). **(Right)** All vertices that lie in $\mathcal{A}_{\mathrm{dom}}(u)$ (red) are dominated by u.

as depicted in Figure 6.2 (left). We define the *dominance area* $\mathcal{A}_{\mathrm{dom}}(u)$ of u to be the area containing all such vertices v. That is,

$$\mathcal{A}_{\mathrm{dom}}(u) = \left\{\mathcal{P} \in \mathcal{D}_R \mid \mathcal{D}_R(\mathcal{P}) \cap \mathcal{D}_R \subseteq \mathcal{D}_R(u)\right\}.$$

The result is illustrated in Figure 6.2 (right). We note that it is sufficient for a vertex v to lie in $\mathcal{A}_{\mathrm{dom}}(u)$ in order to be dominated by u, however, it is not necessary.

Given the radius $r(u)$ of vertex u we can now compute a lower bound on the probability that u dominates another vertex, i.e., the probability that at least one vertex lies in $\mathcal{A}_{\mathrm{dom}}(u)$, by determining the measure $\mu(\mathcal{A}_{\mathrm{dom}}(u))$. To this end, we first define $\delta_{\varphi,\mathrm{dom}}(r(u), r(v))$ to be the maximum angular distance between two vertices u and v such that v lies in $\mathcal{A}_{\mathrm{dom}}(u)$.

Lemma 6.2. *Let u, v be two vertices in $\mathcal{D}_R$. Then, $v \in \mathcal{A}_{\mathrm{dom}}(u)$ if and only if $r(v) \geq r(u)$ and $\delta_\varphi(u,v) \leq \delta_{\varphi,\mathrm{dom}}(r(u), r(v))$, where*

$$\delta_{\varphi,\mathrm{dom}}(r(u), r(v)) = 2(e^{-r(u)/2} - e^{-r(v)/2}) + \Theta\left(e^{-3/2 \cdot r(u)}\right) - \Theta\left(e^{-3/2 \cdot r(v)}\right).$$

Proof. To prove the claim, we consider the possible positions that v can have relative to u and identify the ones for which $v \in \mathcal{A}_{\mathrm{dom}}(u)$ holds.

Assume without loss of generality that $\varphi(u) = 0$, as depicted in Figure 6.3. By definition, $v \in \mathcal{A}_{\mathrm{dom}}(u)$ if and only if $\mathcal{D}_R(v) \cap \mathcal{D}_R \subseteq \mathcal{D}_R(u)$. First note that this

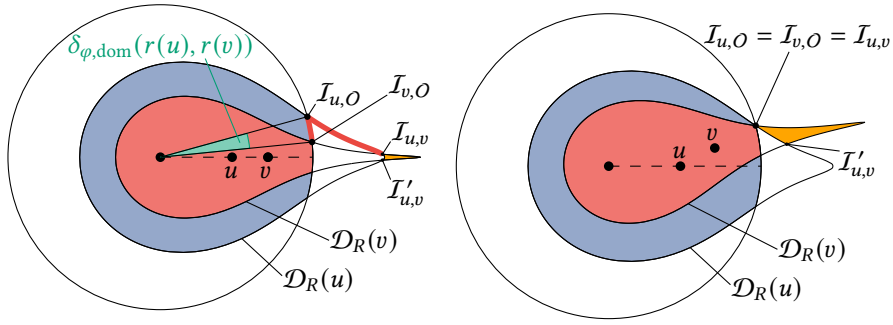


Figure 6.3: (Left) Vertex v is in the dominance area of u, since $\mathcal{D}_R(v) \cap \mathcal{D}_R$ (red area) is contained in $\mathcal{D}_R(u)$. The intersections $\mathcal{I}_{u,v}, \mathcal{I}'_{u,v}$ mark the separation between $\mathcal{D}_R(v) \setminus \mathcal{D}_R(u)$ (orange area) and the rest of $\mathcal{D}_R(v)$. If v is rotated in counterclockwise direction, $\mathcal{I}_{v,O}$ and $\mathcal{I}_{u,v}$ move along the red lines towards $\mathcal{I}_{u,O}$. **(Right)** Vertex v is rotated such that $\mathcal{I}_{u,v} = \mathcal{I}_{u,O}$.

is not the case if $r(v) < r(u)$, as then for the point $\mathcal{P} = (R - r(v), \pi)$ it holds that $\mathcal{P} \in \mathcal{D}_R(v) \cap \mathcal{D}_R$ but $\mathcal{P} \notin \mathcal{D}_R(u)$ for all $\varphi(v) \in [0, 2\pi)$. When $r(v) \geq r(u)$, we know that $\mathcal{D}_R(v) \cap \mathcal{D}_R \subseteq \mathcal{D}_R(u)$ holds when u and v have the same angular coordinate, due to Lemma 3.2. This shows that the first condition ($r(v) \geq r(u)$) is necessary for v to be in the dominance area of u, and it remains to determine the maximum angular deviation between the two, such that this is still the case.

To this end, we argue about intersections of $\mathcal{D}_R(u)$, $\mathcal{D}_R(v)$, and $\mathcal{D}_R$, which we use as indicators whether $v \in \mathcal{A}_{\mathrm{dom}}(u)$ holds. For now assume that $\varphi(v) = \varphi(u)$ and consider the two intersections $\mathcal{I}_{u,v}, \mathcal{I}'_{u,v}$ of $\mathcal{D}_R(u)$ with $\mathcal{D}_R(v)$, as depicted in Figure 6.3 (left). Since $\mathcal{D}_R(v) \cap \mathcal{D}_R \subseteq \mathcal{D}_R(u)$ holds by Lemma 3.2 and since circles are convex, we know that $\mathcal{D}_R(v) \setminus \mathcal{D}_R(u)$ (the orange area in Figure 6.3 (left)) lies outside of $\mathcal{D}_R$ and so do the two intersections $\mathcal{I}_{u,v}, \mathcal{I}'_{u,v}$. For the same reason, we know that $\mathcal{I}_{v,O}$, the intersection of $\mathcal{D}_R(v)$ with $\mathcal{D}_R$ whose angular coordinate is $\varphi(\mathcal{I}_{v,O}) \in [0, \pi]$, lies in $\mathcal{D}_R(u)$. It follows that, for the analogously defined intersection $\mathcal{I}_{u,O}$ we have $\varphi(\mathcal{I}_{v,O}) \leq \varphi(\mathcal{I}_{u,O})$.

We now relax the assumption that $\varphi(v) = \varphi(u)$ and instead imagine that we increase the angle between u and v by some $\delta > 0$, which denotes a counterclockwise rotation of v around the origin. (For symmetry reasons the argumentation about a clockwise rotation is analogous.) Then, $\mathcal{I}_{u,v}$ and $\mathcal{I}'_{u,v}$ move along the boundary of $\mathcal{D}_R(u)$ and, in particular, $\mathcal{I}_{u,v}$ moves towards $\mathcal{I}_{u,O}$. Note that at the same time $\mathcal{I}_{v,O}$ moves towards $\mathcal{I}_{u,O}$ as well. Both movements are depicted

using red lines in Figure 6.3 (left). As long as $\mathcal{I}_{u,v}$ has not surpassed $\mathcal{I}_{u,O}$, neither of the two intersections of $\mathcal{D}_R(v)$ with $\mathcal{D}_R(u)$ lies inside of $\mathcal{D}_R$, which means that $\mathcal{D}_R(v) \setminus \mathcal{D}_R(u)$ remains outside of $\mathcal{D}_R$ and we maintain the property that $\mathcal{D}_R(v) \cap \mathcal{D}_R \subseteq \mathcal{D}_R(u)$. As we keep increasing δ, we eventually get to the point where $\mathcal{I}_{u,v}$ reaches $\mathcal{I}_{u,O}$, as depicted in Figure 6.3 (right). Note that at this point we also have $\mathcal{I}_{v,O} = \mathcal{I}_{u,v}$. Consequently, if we were to rotate v any further, we would have $\mathcal{I}_{v,O} \notin \mathcal{D}_R(u)$, meaning $\mathcal{D}_R(v) \cap \mathcal{D}_R$ would no longer be a subset of $\mathcal{D}_R(u)$. It follows that $\mathcal{D}_R(v) \cap \mathcal{D}_R \subseteq \mathcal{D}_R(u)$ if and only if $\varphi(\mathcal{I}_{v,O}) \leq \varphi(\mathcal{I}_{u,O})$.

To compute the maximum angular distance between u and v such that this is the case, we again start with the assumption that $\varphi(v) = \varphi(u) = 0$, and determine the maximum angle $\delta_{\varphi,\mathrm{dom}}(r(u), r(v))$ such that

$$\varphi(\mathcal{I}_{v,O}) + \delta_{\varphi,\mathrm{dom}}(r(u), r(v)) \leq \varphi(\mathcal{I}_{u,O}).$$

Since $\mathcal{I}_{u,O}$ and $\mathcal{I}_{v,O}$ have radius R and hyperbolic distance R from u and v, respectively, we can apply Equation (3.10) to compute their angular coordinates as $\varphi(\mathcal{I}_{u,O}) = \theta(r(u), R)$ and $\varphi(\mathcal{I}_{v,O}) = \theta(r(v), R)$, respectively. Substituting these angles in the above inequality yields

$$\theta(r(v), R) + \delta_{\varphi,\mathrm{dom}}(r(u), r(v)) \leq \theta(r(u), R).$$

We can now solve for $\delta_{\varphi,\mathrm{dom}}(r(u), r(v))$ and apply Equation (3.10) to obtain

$$\begin{aligned}\delta_{\varphi,\mathrm{dom}}(r(u), r(v)) &= \theta(r(u), R) - \theta(r(v), R) \\ &= 2(e^{-r(u)/2} - e^{-r(v)/2}) + \Theta\left(e^{-3/2 \cdot r(u)}\right) - \Theta\left(e^{-3/2 \cdot r(v)}\right). \qquad \square\end{aligned}$$

Using Lemma 6.2 we can now compute the probability for a given vertex to lie in the dominance area of u. We note that this probability grows roughly like $2/\pi \cdot e^{-r(u)/2}$, which is a constant fraction of the measure of the neighborhood disk of u, which grows as $\alpha/(\alpha - 1/2) \cdot 2/\pi \cdot e^{-r(u)/2}$, see Equation (3.9). Consequently, the expected number of vertices that u dominates at least is a constant fraction of the expected number of its neighbors.

Lemma 6.3. *Let u be a vertex with radius $r(u) \geq R/2$. The probability for a given vertex to lie in $\mathcal{A}_{\mathrm{dom}}(u)$ is given by*

$$\mu\big(\mathcal{A}_{\mathrm{dom}}(u)\big) = \frac{2}{\pi} e^{-r(u)/2}\left(1 - \mathrm{O}\left(e^{-\alpha(R - r(u))}\right)\right) \pm \mathrm{O}\big(1/n\big).$$

Proof. The probability for a given vertex v to lie in $\mathcal{A}_{\text{dom}}(u)$ is obtained by integrating the probability density (given by Equation (3.7)) over $\mathcal{A}_{\text{dom}}(u)$.

$$\begin{aligned}
\mu\big(\mathcal{A}_{\text{dom}}(u)\big) &= 2\int_{r(u)}^{R}\int_{0}^{\delta_{\varphi,\text{dom}}(r(u),r)} f(r,\varphi)\ \mathrm{d}\varphi\,\mathrm{d}r \\
&= 2\int_{r(u)}^{R} \delta_{\varphi,\text{dom}}(r(u),r) f(r)\,\mathrm{d}r \\
&= 2\int_{r(u)}^{R}\left(2\left(e^{-r(u)/2}-e^{-r/2}\right)+\Theta\left(e^{-3/2\cdot r(u)}\right)-\Theta\left(e^{-3/2\cdot r}\right)\right) \\
&\qquad \cdot\frac{\alpha}{2\pi}e^{-\alpha(R-r)}\left(1+\Theta\left(e^{-\alpha R}-e^{-2\alpha r}\right)\right)\mathrm{d}r
\end{aligned}$$

Since $r(u) \geq R/2$ and $r \in [r(u), R]$ we have

$$\Theta\left(e^{-3/2\cdot r(u)}\right)-\Theta\left(e^{-3/2\cdot r}\right)=\pm\mathrm{O}\left(e^{-3/4\cdot R}\right)$$

as well as

$$\left(1+\Theta\left(e^{-\alpha R}-e^{-2\alpha r}\right)\right)=\left(1+\mathrm{O}\left(e^{-\alpha R}\right)\right).$$

Due to the linearity of integration, constant factors within the integrand can be moved out of the integral, which yields

$$\begin{aligned}
\mu\big(\mathcal{A}_{\text{dom}}(u)\big) &= \frac{\alpha}{\pi}e^{-\alpha R}\left(1+\mathrm{O}\left(e^{-\alpha R}\right)\right) \\
&\qquad\cdot\int_{r(u)}^{R}\left(2\left(e^{-r(u)/2}-e^{-r/2}\right)\pm\mathrm{O}\left(e^{-3/4\cdot R}\right)\right)\cdot e^{\alpha r}\,\mathrm{d}r \\
&= \frac{2\alpha}{\pi}e^{-r(u)/2}e^{-\alpha R}\left(1+\mathrm{O}\left(e^{-\alpha R}\right)\right)\int_{r(u)}^{R}e^{\alpha r}\,\mathrm{d}r \\
&\quad -\frac{2\alpha}{\pi}e^{-\alpha R}\left(1+\mathrm{O}\left(e^{-\alpha R}\right)\right)\int_{r(u)}^{R}e^{(\alpha-1/2)r}\,\mathrm{d}r \\
&\quad \pm\mathrm{O}\left(e^{-(3/4+\alpha)R}\int_{r(u)}^{R}e^{\alpha r}\,\mathrm{d}r\right).
\end{aligned}$$

The remaining integrals can be computed easily and we obtain

$$\begin{aligned}\mu\big(\mathcal{A}_{\text{dom}}(u)\big) = \frac{2}{\pi}e^{-r(u)/2}\Big(1+\mathrm{O}\Big(e^{-\alpha R}\Big)\Big)\Big(1-e^{-\alpha(R-r(u))}\Big)\\ - \frac{2\alpha}{(\alpha-1/2)\pi}e^{-R/2}\Big(1+\mathrm{O}\Big(e^{-\alpha R}\Big)\Big)\Big(1-e^{-(\alpha-1/2)(R-r(u))}\Big)\\ \pm \mathrm{O}\Big(e^{-3/4\cdot R}\Big(1-e^{-\alpha(R-r(u))}\Big)\Big).\end{aligned} \tag{6.1}$$

It remains to simplify the remaining error terms. To do this, we consider the three summands in the above expression separately, starting with the first. There, the error term can be expanded to obtain

$$\begin{aligned}&\Big(1+\mathrm{O}\Big(e^{-\alpha R}\Big)\Big)\Big(1-e^{-\alpha(R-r(u))}\Big)\\ &\qquad = 1+\mathrm{O}\Big(e^{-\alpha R}\Big) - e^{-\alpha(R-r(u))} - \mathrm{O}\Big(e^{-\alpha R}\cdot e^{-\alpha(R-r(u))}\Big)\\ &\qquad = 1+e^{-\alpha R}\Big(\mathrm{O}(1) - e^{\alpha r(u)} - \mathrm{O}\Big(e^{-\alpha(R-r(u))}\Big)\Big).\end{aligned}$$

Now recall that $R = 2\log(n)+C$ for a constant $C\in\mathbb{R}$ (see Section 3.3). Moreover, since $r(u)\geq R/2$ holds by assumption, we have $e^{\alpha r(u)} = \omega(1)$ and thus $\mathrm{O}(1) - e^{\alpha r(u)} = -\mathrm{O}\Big(e^{\alpha r(u)}\Big)$. We obtain

$$\Big(1+\mathrm{O}\Big(e^{-\alpha R}\Big)\Big)\Big(1-e^{-\alpha(R-r(u))}\Big) = 1+e^{-\alpha R}\Big(-\mathrm{O}\Big(e^{\alpha r(u)}\Big) - \mathrm{O}\Big(e^{-\alpha(R-r(u))}\Big)\Big).$$

Again, since $R = 2\log(n)+C$ for a constant C, we have $e^{-\alpha R} = \mathrm{o}(1)$ and thus $\mathrm{O}\big(e^{-\alpha(R-r(u))}\big) = \mathrm{O}\big(e^{\alpha r(u)}\big)$. Therefore, the error term further simplifies to $\big(1-\mathrm{O}\big(e^{-\alpha(R-r(u))}\big)\big)$ and Equation (6.1) becomes

$$\begin{aligned}\mu\big(\mathcal{A}_{\text{dom}}(u)\big) = \frac{2}{\pi}e^{-r(u)/2}\Big(1-\mathrm{O}\Big(e^{-\alpha(R-r(u))}\Big)\Big)\\ - \frac{2\alpha}{(\alpha-1/2)\pi}e^{-R/2}\Big(1+\mathrm{O}\Big(e^{-\alpha R}\Big)\Big)\Big(1-e^{-(\alpha-1/2)(R-r(u))}\Big)\\ \pm \mathrm{O}\Big(e^{-3/4\cdot R}\Big(1-e^{-\alpha(R-r(u))}\Big)\Big).\end{aligned}$$

Now consider the second summand. Since α is constant, so is the first fraction. Moreover, as $R = 2\log(n) + C$ for a constant C, we have

$$\left(1 + \mathrm{O}\left(e^{-\alpha R}\right)\right) = \left(1 + \mathrm{o}(1)\right) = \mathrm{O}(1).$$

And since $r(u) \leq R$, the exponent in the last factor is non-positive, from which we can conclude that this factor is also $\mathrm{O}(1)$. The second summand therefore simplifies to $\mathrm{O}\left(e^{-R/2}\right) = \mathrm{O}\left(n^{-1}\right)$. Finally, the last summand can be reduced to $\mathrm{O}\left(e^{-3/4 \cdot R}\right) = \mathrm{O}\left(n^{-3/2}\right)$, which yields

$$\mu\left(\mathcal{A}_{\mathrm{dom}}(u)\right) = \frac{2}{\pi} e^{-r(u)/2}\left(1 - \mathrm{O}\left(e^{-\alpha(R-r(u))}\right)\right) - \mathrm{O}\left(n^{-1}\right) \pm \mathrm{O}\left(n^{-3/2}\right).$$

Combining the last two summands then yields the claim. □

The following lemma shows that, with high probability, all vertices that are not too close to the boundary of the disk dominate at least one vertex.

Lemma 6.4. *Let G be a hyperbolic random graph on n vertices with power-law exponent $2\alpha + 1$ and expected average degree κ. Then, there is a constant $c > 2/(\kappa(1 - 1/(2\alpha))^2)$, such that all vertices u with $r(u) \leq \rho = R - 2\log\log(n^c)$ are dominant, with high probability.*

Proof. Vertex u is dominant if at least one vertex lies in $\mathcal{A}_{\mathrm{dom}}(u)$. To show this for any u with $r(u) \leq \rho$, it suffices to show it for $r(u) = \rho$, since $\mu(\mathcal{A}_{\mathrm{dom}}(u))$ increases with decreasing radius. To determine the probability that at least one vertex lies in $\mathcal{A}_{\mathrm{dom}}(u)$, we use Lemma 6.3 and obtain

$$\begin{aligned}\mu\left(\mathcal{A}_{\mathrm{dom}}(u)\right) &= \frac{2}{\pi} e^{-\rho/2}\left(1 - \mathrm{O}\left(e^{-\alpha(R-\rho)}\right)\right) \pm \mathrm{O}\left(1/n\right) \\ &= \frac{2}{\pi} e^{-R/2 + \log\log(n^c)}\left(1 - \mathrm{O}\left(e^{-2\alpha\log\log(n^c)}\right)\right) \pm \mathrm{O}\left(1/n\right).\end{aligned}$$

By substituting the value for R (see Section 3.3), which is

$$R = 2\log\left(\frac{2n}{\pi\kappa} \cdot \left(\frac{\alpha}{\alpha - 1/2}\right)^2 \left(1 + \mathrm{o}(1)\right)\right),$$

we obtain

$$\mu\big(\mathcal{A}_{\text{dom}}(u)\big) = \frac{\kappa}{n}\left(\frac{\alpha - 1/2}{\alpha}\right)^2 \frac{1}{1+\mathrm{o}(1)} c\log(n)\left(1 - \mathrm{O}\Big(\log(n)^{-2\alpha}\Big)\right) \pm \mathrm{O}\big(1/n\big).$$

Moreover, since $1/(1+x) = 1 - \Theta(x)$ for $x \in \mathbb{R}$ with $x = \pm\mathrm{o}(1)$ (Lemma 2.11), we can conclude that

$$\mu\big(\mathcal{A}_{\text{dom}}(u)\big) = c\kappa\left(1 - \frac{1}{2\alpha}\right)^2 \frac{\log(n)}{n}\big(1 - \mathrm{o}(1)\big) \pm \mathrm{O}\big(1/n\big).$$

The probability of at least one vertex falling into $\mathcal{A}_{\text{dom}}(u)$ is now given by

$$\begin{aligned}\Pr\big[V\big(\mathcal{A}_{\text{dom}}(u)\big) \neq \emptyset\big] &= 1 - \Big(1 - \mu\big(\mathcal{A}_{\text{dom}}(u)\big)\Big)^n \\ &\geq 1 - e^{-n\mu\left(\mathcal{A}_{\text{dom}}(u)\right)} \\ &= 1 - \Theta\Big(n^{-c\kappa(1-1/(2\alpha))^2(1-\mathrm{o}(1))}\Big),\end{aligned}$$

where the second inequality is due to the fact that $1 + x \leq e^x$ for all $x \in \mathbb{R}$ (see Lemma 2.9). As a consequence, for large enough n we can choose a constant $c > 2/(\kappa(1-1/(2\alpha))^2)$, such that the probability of a vertex at radius ρ being dominant is at least $1 - \Theta\big(n^{-2}\big)$, allowing us to apply the union bound. □

Corollary 6.5. *Let G be a hyperbolic random graph on n vertices with power-law exponent $2\alpha + 1$ and expected average degree κ. Then, there exists a constant $c > 2/(\kappa(1-1/(2\alpha))^2)$, such that all vertices with radius at most $\rho = R - 2\log\log(n^c)$ are removed by the dominance rule, with high probability.*

6.2.2 Simple Structure in the Outer Band

By Corollary 6.5 the dominance rule removes all vertices of radius at most ρ. Consequently, all remaining vertices have radius at least ρ. We refer to this part of the disk as *outer band*. More precisely, the outer band is defined as $\mathcal{D}_R \setminus \mathcal{D}_\rho$.

To show that we can solve the vertex cover problem efficiently on this part of the graph, we utilize the fact that it has a simple structure. More precisely, we prove that this remainder has a small *pathwidth*, which is a parameter related to tree decompositions. A *tree decomposition* of a graph G is a tree T where each vertex in the tree represents a subset of the vertices of G called a *bag*, and the fol-

lowing requirements have to be satisfied: Each vertex in G is contained in at least one bag, all bags containing a given vertex in G form a connected subtree of T, and for each edge in G, there exists a bag containing both endpoints. The *width* of a tree decomposition is the size of its largest bag minus one. The *treewidth* of G is the minimum width over all tree decompositions of G. In our analysis, we consider special tree decompositions called *path decomposition*, which are defined analogously, with the added constraint that the tree has to be a path. Additionally, the *pathwidth* of a graph G, denoted by $\mathrm{pw}(G)$, is the minimum width over all path decompositions of G. In the following, we prove that the remaining graph in the outer band has a small pathwidth. Since the pathwidth is an upper bound on the treewidth, we can utilize existing results stating that, in that case, tree decompositions can be computed efficiently [Cyg+15, Theorem 7.18], which in turn allows us to solve the vertex cover problem, efficiently [Cyg+15, Theorem 7.9].

The idea now is to utilize the underlying geometry to find a *circular arc super graph* of the remainder, for which the pathwidth can be bounded easily. Circular arc graphs are a super class of interval graphs. In an *interval graph* each vertex v is identified with an interval on the real line and two vertices are adjacent if and only if their intervals intersect. The *interval width* of an interval graph G, denoted by $\mathrm{iw}(G)$, is its maximum clique size, i.e., the maximum number of intervals that intersect in one point. For any graph the interval width is defined as the minimum interval width over all of its interval supergraphs. In *circular arc graphs* each vertex is identified with a subinterval of the circle called *circular arc* or simply *arc*. The interval width of a circular arc graph G is at most twice the size of its maximum clique, since one obtains an interval supergraph of G by mapping the circular arcs into the interval $[0, 2\pi]$ on the real line and replacing all intervals that were split by this mapping with the whole interval $[0, 2\pi]$. Consequently, for any graph G, if k denotes the minimum over the maximum clique number of all circular arc supergraphs G' of G, then the interval width of G is at most $2k$. It is known that for any graph G and any $k \geq 0$, the interval width of G is at most $k + 1$ if and only if its pathwidth is at most k [Cyg+15, Theorem 7.14]. Consequently, if k' is the maximum clique size of a circular arc supergraph of G, then $2k' - 1$ is an upper bound on the pathwidth of G.

In the following, we use $G|_{r(v)\geq r} = G[\{v \in V\} \mid r(v) \geq r]$ to denote the induced subgraph of G that contains all vertices with radius at least r. To show that the pathwidth of $G|_{r(v)\geq \rho}$ (the induced subgraph in the outer band) is small,

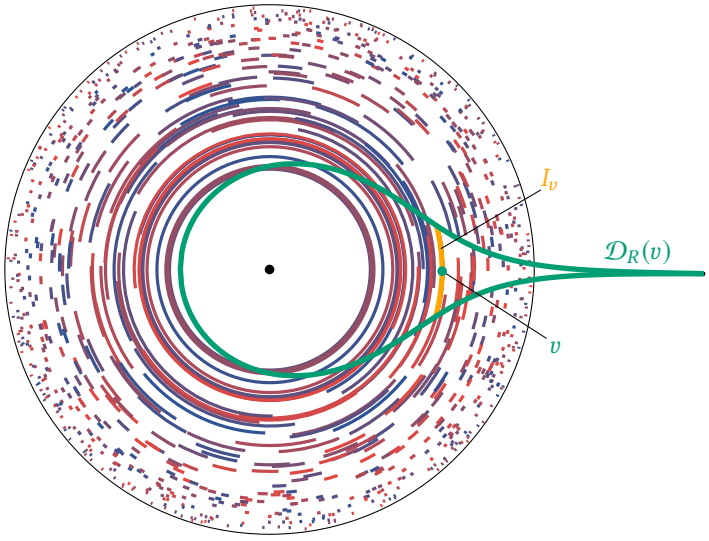


Figure 6.4: The angular intervals representing the circular arc supergraph $\hat{G}$ of a hyperbolic random graph G. The arc I_v (orange) of a vertex v extends to the boundary of its neighborhood disk $\mathcal{D}_R(v)$ (green) at the radius of v.

we first show that there is a circular arc supergraph $\hat{G}|_{r(v)\geq\rho}$ of $G|_{r(v)\geq\rho}$ with a small maximum clique. We use $\hat{G}$ to denote a circular arc supergraph of a hyperbolic random graph G, which is obtained by assigning each vertex v an angular interval I_v on the circle, such that the intervals of two adjacent vertices intersect. More precisely, for a vertex v, we set

$$I_v = [\varphi(v) - \theta(r(v), r(v)), \varphi(v) + \theta(r(v), r(v))],$$

where $\theta(r(u), r(v))$ denotes the maximum angular distance such that to vertices u and v are adjacent, see Section 3.3.5. Intuitively, this means that the interval of a vertex contains a superset of all its neighbors with larger radius, as can be seen in Figure 6.4. The following lemma shows that $\hat{G}$ is actually a supergraph of G.

Lemma 6.6. *Let $G = (V, E)$ be a hyperbolic random graph. Then, the graph $\hat{G}$ is a supergraph of G.*

Proof. Let $\{u, v\} \in E$ be any edge in G. To show that $\hat{G}$ is a supergraph of G we need to show that u and v are also adjacent in $\hat{G}$, i.e., $I_u \cap I_v \neq \emptyset$. Without

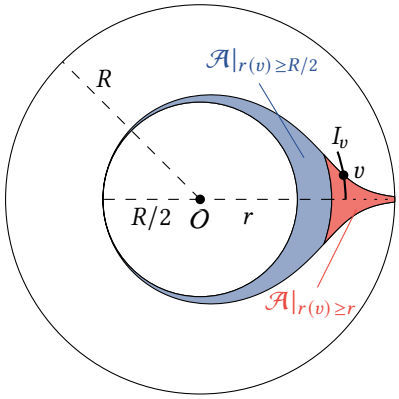


Figure 6.5: The area that contains the vertices whose arcs intersect angle 0. Area $\mathcal{A}|_{r(v)\geq r}$ (red) contains all such vertices with radius at least r. Vertex v lies on the boundary of $\mathcal{A}|_{r(v)\geq r}$ and its interval I_v extends to 0.

loss of generality assume $r(u) \leq r(v)$. Since u and v are adjacent in G, the hyperbolic distance between them is at most R. It follows, that their angular distance $\delta_\varphi(u,v)$ is bounded by $\theta(r(u), r(v))$. Since $\theta(r(u), r(v)) \leq \theta(r(u), r(u))$ for $r(u) \leq r(v)$, we have $\delta_\varphi(u,v) \leq \theta(r(u), r(u))$. As I_u extends by $\theta(r(u), r(u))$ from $\varphi(u)$ in both directions, it follows that $\varphi(v) \in I_u$. □

Note that $\hat{G}$ is still a supergraph of G, after removing a vertex from both G and $\hat{G}$. Consequently, $\hat{G}|_{r(v)\geq\rho}$ is a supergraph of $G|_{r(v)\geq\rho}$. It remains to show that $\hat{G}|_{r(v)\geq\rho}$ has a small maximum clique number, which is given by the maximum number of arcs that intersect at any angle. To this end, we first compute this number at a given angle, which we set to 0 without loss of generality. Let $\mathcal{A}|_{r(v)\geq r}$ denote the area of the disk containing all vertices v with radius $r(v) \geq r$ whose interval I_v intersects 0, as illustrated in Figure 6.5. The following lemma describes the probability for a given vertex to lie in $\mathcal{A}|_{r(v)\geq r}$.

Lemma 6.7. *Let G be a hyperbolic random graph and let $r \geq R/2$. The probability for a given vertex to lie in $\mathcal{A}|_{r(v)\geq r}$ is bounded by*

$$\mu\left(\mathcal{A}|_{r(v)\geq r}\right) \leq \frac{2\alpha}{(1-\alpha)\pi} e^{-(\alpha-1/2)R-(1-\alpha)r} \cdot \left(1 + \mathrm{O}\left(e^{-\alpha R} + e^{-(2r-R)}\right) - \mathrm{O}\left(e^{-(1-\alpha)(R-r)}\right)\right).$$

Proof. We obtain the measure of $\mathcal{A}|_{r(v)\geq r}$ by integrating the probability density function over $\mathcal{A}|_{r(v)\geq r}$, see Section 3.3. Due to the definition of I_v we can conclude that $\mathcal{A}|_{r(v)\geq r}$ includes all vertices v with radius $r(v) \geq r$ whose angular distance to 0 is at most $\theta(r(v), r(v))$, defined in Equation (3.10). We obtain,

$$\begin{aligned}\mu\Big(\mathcal{A}|_{r(v)\geq r}\Big) &= \int_r^R 2\int_0^{\theta(x,x)} f(x,\varphi)\,\mathrm{d}\varphi\,\mathrm{d}x \\ &= 2\int_r^R \left(2e^{(R-2x)/2}\left(1 \pm \Theta\left(e^{R-2x}\right)\right)\right. \\ &\qquad \left.\cdot \frac{\alpha}{2\pi}e^{-\alpha(R-x)}\left(1+\Theta\left(e^{-\alpha R}-e^{-2\alpha x}\right)\right)\right)\mathrm{d}x.\end{aligned}$$

As before, we can conclude that $\left(1+\Theta\left(e^{-\alpha R}-e^{-2\alpha r}\right)\right) = \left(1+\mathrm{O}\left(e^{-\alpha R}\right)\right)$, since $r \geq R/2$. By moving constant factors out of the integral, the expression can be simplified to

$$\mu\Big(\mathcal{A}|_{r(v)\geq r}\Big) \leq \frac{2\alpha}{\pi}e^{-(\alpha-1/2)R}\left(1+\mathrm{O}\left(e^{-\alpha R}\right)\right)\int_r^R e^{-(1-\alpha)x}\left(1+\Theta\left(e^{R-2x}\right)\right)\mathrm{d}x.$$

We split the sum in the integral and deal with the resulting integrals separately.

$$\begin{aligned}\mu\Big(\mathcal{A}|_{r(v)\geq r}\Big) &\leq \frac{2\alpha}{\pi}e^{-(\alpha-1/2)R}\left(1+\mathrm{O}\left(e^{-\alpha R}\right)\right) \\ &\qquad \cdot\left(\int_r^R e^{-(1-\alpha)x}\,\mathrm{d}x + \Theta\left(\int_r^R e^{-(1-\alpha)x+R-2x}\,\mathrm{d}x\right)\right) \\ &= \frac{2\alpha}{\pi}e^{-(\alpha-1/2)R}\left(1+\mathrm{O}\left(e^{-\alpha R}\right)\right) \\ &\qquad \cdot\left(\frac{1}{1-\alpha}e^{-(1-\alpha)r}\left(1-e^{-(1-\alpha)(R-r)}\right)\right. \\ &\qquad \left.+\Theta\left(e^{R}e^{-(3-\alpha)r}\left(1-e^{-(3-\alpha)(R-r)}\right)\right)\right).\end{aligned}$$

By placing $1/(1-\alpha)\cdot e^{-(1-\alpha)r}$ outside of the parentheses, we obtain

$$\mu\Big(\mathcal{A}|_{r(v)\geq r}\Big) \leq \frac{2\alpha}{(1-\alpha)\pi} e^{-(\alpha-1/2)R-(1-\alpha)r}\Big(1+\mathrm{O}\Big(e^{-\alpha R}\Big)\Big) \cdot \left(\Big(1-e^{-(1-\alpha)(R-r)}\Big)+\Theta\Big(e^{R-2r}\Big(1-e^{-(3-\alpha)(R-r)}\Big)\Big)\right).$$

Simplifying the remaining error terms then yields the claim. □

We can now bound the maximum clique number in $\hat{G}|_{r(v)\geq\rho}$ and with that its interval width $\mathrm{iw}\big(\hat{G}|_{r(v)\geq\rho}\big)$.

Theorem 6.8. *Let G be a hyperbolic random graph on n vertices and let $r \geq R/2$ be a radius. Then, there exists a constant c such that, with high probability, it holds that $\mathrm{iw}\big(\hat{G}|_{r(v)\geq r}\big) = \mathrm{O}\big(\log(n)\big)$, if $r \geq R - 1/(1-\alpha)\cdot\log\log(n^c)$, and otherwise*

$$\mathrm{iw}\big(\hat{G}|_{r(v)\geq r}\big) \leq \frac{5\alpha}{(1-\alpha)\pi} n e^{-(\alpha-1/2)R-(1-\alpha)r} \cdot \left(1+\mathrm{O}\Big(e^{-\alpha R}+e^{-(2r-R)}\Big)-\mathrm{O}\Big(e^{-(1-\alpha)(R-r)}\Big)\right).$$

Proof. We start by determining the expected number of arcs that intersect at a given angle. This can be done by computing the expected number of vertices in $\mathcal{A}|_{r(v)\geq r}$ using Lemma 6.7, which yields.

$$\mathbb{E}\Big[\big|V\big(\mathcal{A}|_{r(v)\geq r}\big)\big|\Big] \leq \frac{2\alpha}{(1-\alpha)\pi} n e^{-(\alpha-1/2)R-(1-\alpha)r} \cdot \left(1+\mathrm{O}\Big(e^{-\alpha R}+e^{-(2r-R)}\Big)-\mathrm{O}\Big(e^{-(1-\alpha)(R-r)}\Big)\right).$$

We denote this upper bound with $g(r)$. It remains to show that it also holds with high probability at every angle. To this end, we apply a Chernoff bound (Corollary 2.2) to conclude that for any $\varepsilon \in (0,1)$ it holds that

$$\Pr\Big[\big|V\big(\mathcal{A}|_{r(v)\geq r}\big)\big| \geq (1+\varepsilon)g(r)\Big] \leq e^{-\varepsilon^2/3\cdot g(r)}.$$

In order to see that this probability is sufficiently small, we first take a closer look at $g(r')$ with $r' = R - 1/(1-\alpha)\cdot\log\log(n^c)$ and afterwards argue about

the different values that r can take relative to r'.

$$\begin{aligned} g(r') &= \frac{2\alpha}{(1-\alpha)\pi} n e^{-(\alpha-1/2)R-(1-\alpha)(R-1/(1-\alpha)\cdot\log\log(n^c))} \\ &\quad \cdot \left(1 + \mathrm{O}\left(e^{-\alpha R} + e^{-(2(R-1/(1-\alpha)\cdot\log\log(n^c))-R)}\right)\right. \\ &\quad \left. - \mathrm{O}\left(e^{-(1-\alpha)(R-(R-1/(1-\alpha)\log\log(n^c)))}\right)\right) \\ &= \frac{2\alpha}{(1-\alpha)\pi} n e^{-R/2+\log\log(n^c)} \\ &\quad \cdot \left(1 + \mathrm{O}\left(e^{-\alpha R} + e^{-(R-2/(1-\alpha)\cdot\log\log(n^c))}\right) - \mathrm{O}\left(e^{-\log\log(n^c)}\right)\right) \end{aligned}$$

Substituting the value for R as defined in Section 3.3.1, which is

$$R = 2\log\left(\frac{2n}{\pi\kappa}\cdot\left(\frac{\alpha}{\alpha-1/2}\right)^2(1+\mathrm{o}(1))\right),$$

we obtain

$$g(r') = c\kappa\frac{(\alpha-1/2)^2}{(1-\alpha)\alpha}\log(n)(1\pm o(1)).$$

Now consider the case where $r < r'$. Then, $g(r) > g(r')$ and applying Corollary 2.2 with $\varepsilon = 1/4$ yields

$$\Pr\left[\left|V\left(\mathcal{A}|_{r(v)\geq r}\right)\right| \geq 5/4\cdot g(r)\right] \leq e^{-\frac{\varepsilon^2}{3}g(r)} \leq e^{-\frac{1}{48}g(r')} \leq n^{-c\kappa\frac{(\alpha-1/2)^2}{48(1-\alpha)\alpha}(1\pm\mathrm{o}(1))}.$$

For the case, where $r \geq r'$, note that $\mathbb{E}\left[|V(\mathcal{A}|_{r(v)\geq r})|\right]$ decreases with increasing r. Therefore, $g(r') = \mathrm{O}(\log(n))$ is a pessimistic but valid upper bound on $g(r)$ and we obtain the same bound on $\Pr\left[|V(\mathcal{A}|_{r(v)\geq r})| \geq 5/4\cdot g(r')\right]$.

In both cases, we can choose c such that $|V(\mathcal{A}|_{r(v)\geq r})| \leq 5/4\cdot g(r)$ holds with probability $1-\mathrm{O}(n^{-c'})$ for any c' at a given angle. In order to see that it holds at every angle, note that it suffices to show that it holds at all arc endings as the number of intersecting arcs does not change in between arc endings. Since there are exactly $2n$ arc endings, we can apply the union bound and obtain that the bound holds with probability $1-\mathrm{O}(n^{-c'+1})$ for any c' at every angle. Since $g(r)$

is an upper bound on the maximum clique size of $\hat{G}|_{r(v)\geq r}$, the interval width of $\hat{G}|_{r(v)\geq r}$ is at most twice as large, as argued above. □

Since the interval width of a circular arc supergraph of G is an upper bound on the pathwidth of G [Cyg+15, Theorem 7.14] and since $\rho \geq R - 1/(1-\alpha) \cdot \log\log(n^c)$ for $\alpha \in (1/2, 1)$, we immediately obtain the following corollary.

Corollary 6.9. *Let G be a hyperbolic random graph on n vertices and let $G|_{r(v)\geq\rho}$ be the subgraph obtained by removing all vertices with radius at most $\rho = R - 2\log\log(n^c)$. Then, with high probability it holds that*

$$\mathrm{pw}(G|_{r(v)\geq\rho}) = \mathrm{O}\big(\log(n)\big).$$

We are now ready to prove our main theorem, which we restate for the sake of readability.

Theorem 6.1. *Let G be a hyperbolic random graph on n vertices. Then the vertex cover problem on G can be solved in $\mathrm{poly}(n)$ time, with high probability.*

Proof. Consider the following algorithm that finds a minimum vertex cover of G. We start with an empty vertex cover S. Initially, all dominant vertices are added to S, which is correct due to the dominance rule. By Lemma 6.4, this includes all vertices of radius at most $\rho = R - 2\log\log(n^c)$, for some constant c, with high probability. Obviously, finding all vertices that are dominant can be done in $\mathrm{poly}(n)$ time. It remains to determine a vertex cover of $G|_{r(v)\geq\rho}$. By Corollary 6.9, the pathwidth of $G|_{r(v)\geq\rho}$ is $\mathrm{O}\big(\log(n)\big)$, with high probability. Since the pathwidth is an upper bound on the treewidth, we can find a tree decomposition of $G|_{r(v)\geq\rho}$ and solve the vertex cover problem in $G|_{r(v)\geq\rho}$ in $\mathrm{poly}(n)$ time [Cyg+15, Theorems 7.18 and 7.9]. □

Moreover, linking the radius of a vertex in Theorem 6.8 with its expected degree leads to the following corollary, which is interesting in its own right. It links the pathwidth to the degree d in the graph $G|_{\deg(v)\leq d} = G[\{v \in V \mid \deg(v) \leq d\}]$, i.e., the subgraph of G induced by vertices of degree at most d.

Corollary 6.10. *Let G be a hyperbolic random graph and let $d \leq \sqrt{n}$. Then, with high probability,* $\mathrm{pw}(G|_{\deg(v)\leq d}) = \mathrm{O}\big(d^{2-2\alpha} + \log(n)\big)$.

Proof. Consider the radius $r = R - 2\log(\xi d)$ for some constant $\xi > 0$, and the graph $G|_{r(v)\geq r}$ that is obtained by removing all vertices of radius at most r. In the

following, we show that $G|_{r(v)\geq r}$ is a supergraph of $G|_{\deg(v)\leq d}$ for large enough ξ. Afterwards, we bound the pathwidth of $G|_{r(v)\geq r}$.

As established in Section 3.3, the expected degree of a vertex with radius r is given by

$$\mathbb{E}\big[\deg(v) \,\big|\, r(v) = r\big] = \frac{2\alpha}{(\alpha-1/2)\pi} n e^{-r/2}\Big(1 \pm \mathrm{O}\Big(e^{-(\alpha-1/2)r}\Big)\Big),$$

see Equation (3.9). By substituting $r = R - 2\log(\xi d)$ together with the expression for R, which is given by

$$R = 2\log\left(\frac{2n}{\pi\kappa}\cdot\left(\frac{\alpha}{\alpha-1/2}\right)^2\big(1+\mathrm{o}(1)\big)\right),$$

we obtain

$$\begin{aligned}
\mathbb{E}\big[\deg(v) \,\big|\, r(v) = r\big] &= \frac{2\alpha}{(\alpha-1/2)\pi} n e^{-R/2+\log(\xi d)} \\
&\quad\cdot\left(1 \pm \mathrm{O}\Big(e^{-(\alpha-1/2)(R-2\log(\xi d))}\Big)\right) \\
&= \frac{2\alpha\kappa}{2(\alpha-1/2)}\left(\frac{\alpha-1/2}{\alpha}\right)^2 \frac{1}{1+\mathrm{o}(1)}\xi\cdot d \\
&\quad\cdot\left(1 \pm \mathrm{O}\Big((d/n)^{(2\alpha-1)}\Big)\right) \\
&= \xi\kappa(1-1/(2\alpha))\cdot d\big(1\pm\mathrm{o}(1)\big).
\end{aligned}$$

Note that for large enough n we can choose ξ sufficiently large, such that

$$\Pr\big[\deg(v)\leq d \,\big|\, r(v)=r\big] \leq \Pr\Big[\deg(v)\leq (1-\varepsilon)\mathbb{E}\big[\deg(v)\,\big|\, r(v)=r\big]\Big],$$

for any $\varepsilon\in(0,1)$. This allows us to apply the third inequality in the Chernoff bound in Theorem 2.1 to conclude that

$$\Pr\big[\deg(v)\leq d \,\big|\, r(v)=r\big] \leq \exp\Big(-\varepsilon^2/2\cdot\xi k(1-1/(2\alpha))\cdot d\big(1\pm\mathrm{o}(1)\big)\Big).$$

First assume that $d\geq\log(n)^{1/(2-2\alpha)}$. We handle the other case later. Note that $1/(2-2\alpha)>1$ for $\alpha\in(1/2,1)$ and, thus, $d\geq\log(n)$. Therefore, we can choose n

and ξ sufficiently large, such that

$$\begin{aligned}\Pr\big[\deg(v) \leq d \,\big|\, r(v) = r\,\big] &\leq n^{-\frac{\varepsilon^2}{2}\xi k(1-1/(2\alpha))\left(1\pm o(1)\right)} \\ &\leq n^{-2}.\end{aligned}$$

Since by Corollary 3.3 smaller radius implies larger expected degree, we can derive the same bound for a given vertex of radius *at most* r. By applying the union bound it follows that, with high probability, no vertex with radius at most r has degree less than or equal to d. Conversely, all vertices with degree at most d have radius at least r. Consequently, $G|_{r(v)\geq r}$ is a supergraph of $G|_{\deg(v)\leq d}$.

To prove the claim, it remains to bound the pathwidth of $G|_{r(v)\geq r}$. When $r > R - 1/(1-\alpha)\cdot\log\log(n^c)$, we can apply the first part of Theorem 6.8 in order to conclude that $\operatorname{iw}(\hat{G}|_{r(v)\geq r}) = O\big(\log(n)\big)$. Otherwise, we use the second part of the theorem, allowing us to derive that the interval width of $G|_{r(v)\geq r}$ is at most

$$\begin{aligned}\operatorname{iw}\Big(\hat{G}|_{r(v)\geq r}\Big) &\leq \frac{5\alpha}{(1-\alpha)\pi} n e^{-(\alpha-1/2)R-(1-\alpha)r} \\ &\qquad\cdot\left(1 + O\Big(e^{-\alpha R} + e^{-(2r-R)}\Big) - O\Big(e^{-(1-\alpha)(R-r)}\Big)\right) \\ &= \frac{5\kappa\alpha\xi^{2-2\alpha}}{2(1-\alpha)}\left(\frac{\alpha-1/2}{\alpha}\right)^2 \frac{1}{(1+o(1))} \\ &\qquad\cdot\left(1 + O\Big(n^{-2\alpha} + (d^2/n)^2\Big) - O\Big(d^{-(2-2\alpha)}\Big)\right) \\ &= \frac{5\kappa(\alpha-1/2)^2\xi^{2-2\alpha}}{2(1-\alpha)\alpha} d^{2-2\alpha}\left(1 \pm O(1)\right) \\ &= O\Big(d^{2-2\alpha}\Big).\end{aligned}$$

As argued above, the interval width is an upper bound on the pathwidth.

For the case where $d < \log(n)^{1/(2-2\alpha)}$ (which we excluded above), consider $G|_{\deg(v)\leq d'}$ for $d' = \log(n)^{1/(2-2\alpha)} > d$. As we already proved the corollary for d', we obtain $\operatorname{pw}(G|_{\deg(v)\leq d'}) = O\big(d'^{2-2\alpha} + \log(n)\big) = O\big(\log(n)\big)$. Since $G|_{\deg(v)\leq d}$ is a subgraph of $G|_{\deg(v)\leq d'}$, the same bound holds for $G|_{\deg(v)\leq d}$. □

6.3 Empirical Evaluation

Our results show that a heterogeneous degree distribution as well as high clustering make the dominance rule very effective. This matches the behavior for real-world networks, which typically exhibit these two properties. However, our analysis actually makes more specific predictions: (I) vertices with sufficiently high degree usually have at least one neighbor they dominate and can thus safely be included in the vertex cover; and (II) the graph remaining after deleting the high-degree vertices has simple structure, i.e., small pathwidth.

To see whether this matches the real world, we ran experiments on 59 networks from several network datasets [Are+; BM06; Kun13; LK14; RA15]. Out of the 59 instances, we can solve the vertex cover problem for 47 networks in reasonable time. We refer to these as *easy*, while the remaining 12 are called *hard*. Note that our theoretical analysis aims at explaining why the easy instances are easy.

Recall from Lemma 6.4 that all vertices with radius at most $R - 2\log\log(n^c)$, with $c > 2/(\kappa(1-1/(2\alpha))^2)$, probably dominate. This corresponds to an expected degree of $2\alpha/(\alpha-1/2)\cdot\log(n)$. Figure 6.6 shows the percentage of dominant vertices among the ones above this degree, for the considered real-world networks. For more than 66 % of the 59 networks, more than 75 % of these vertices were in fact dominant (red and blue). For more than 40 % of the networks, more than 95 % were dominant (blue). Restricted to the 47 easy instances, these increase to 82 % and 51 % of networks, respectively.

Experiments concerning the pathwidth of the resulting graph are much more difficult, due to the lack of efficient tools. Therefore, we used the tool by Tamaki et al. [Tam+17] to heuristically compute upper bounds on the treewidth instead. As in our analysis, we only removed vertices that dominate in the original graph instead of applying the reduction rule exhaustively. On the resulting subgraphs, the treewidth heuristic ran with a 15 min timeout. The resulting treewidth is at most 50 for 44 % of the networks and at most 5 for 25 %, see Figure 6.7. Restricted to easy instances, the values increase to 55 % and 32 %, respectively. Note how on most graphs where almost all high-degree vertices are dominant (blue), we obtained the smallest treewidths. This indicates, that on networks where our first prediction was fulfilled, so was the second one.

While hyperbolic random graphs are clearly an idealized representation of real-world graphs, these experiments indicate that the predictions derived from the model match the real world, at least for a significant fraction of networks.

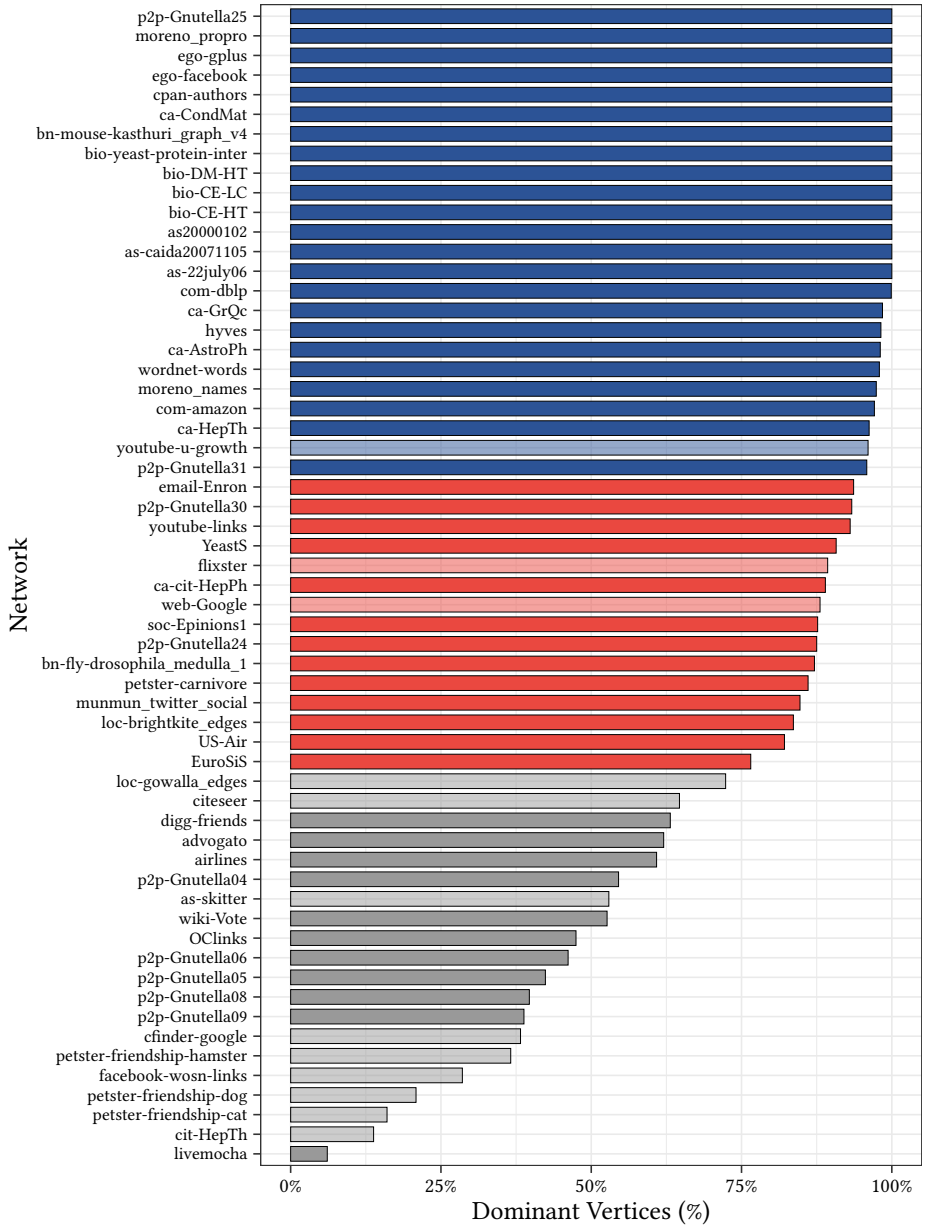


Figure 6.6: Percentage of dominant vertices among ones with degree above $2\alpha/(\alpha - 1/2)\log(n)$. Red bars denote networks where this value is above 75%. Blue bars denote networks where it is above 95%. Transparent bars denote hard instances.

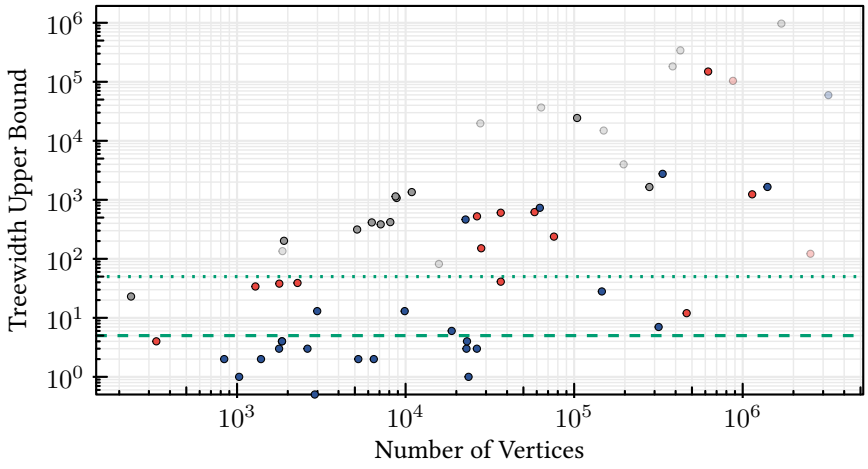


Figure 6.7: Upper bounds on the treewidth of the considered graphs, after removing initially dominant vertices. Dashed and dotted green lines denote a bound of 5 and 50, respectively. Colors represent the percentage of initially dominant high-degree vertices, analogous to Figure 6.6. Transparent dots represent hard instances.

7 Approximate Vertex Cover in Hyperbolic Random Graphs

This chapter is based on joint work together with Thomas Bläsius as well as Tobias Friedrich [BFK21a].

7.1 Introduction

As established in the previous chapter, finding a minimum vertex cover is a fundamental NP-complete graph problem with several relevant applications, which is why there is an ongoing effort in exploring methods that can be used in practice [ACL12; AI16]. While they often work well, they still cannot guarantee efficient running times. This includes the previously shown polynomial running time bound (Chapter 6). There the degree of the polynomial is unknown and on large networks even quadratic algorithms are not efficient enough to obtain results in a reasonable amount of time.

A commonly used approach to overcoming this issue are approximation algorithms. There, the idea is to settle for a *near*-optimal solution while guaranteeing an efficient running time. For the vertex cover problem, a simple greedy approach computes an approximation in quasi-linear time by iteratively adding the vertex with the largest degree to the cover and removing it from the graph. In general graphs, this algorithm, which we call *standard greedy*, cannot beat an approximation ratio of $\Omega(\log(n))$, i.e., there are graphs where it produces a vertex cover whose size exceeds the one of an optimum by a factor of $\Omega(\log(n))$ [Joh74]. This can be improved to a 2-approximation using a simple linear-time algorithm. The best known polynomial time approximation reduces the factor to $2 - \Theta(\log(n)^{-1/2})$ [Kar09]. However, there is reason to believe that it is NP-hard to approximate an optimal vertex cover within a factor of $2-\varepsilon$ for all $\varepsilon > 0$ [KR08] and it is proven that finding a $\sqrt{2}$-approximation is NP-hard [SMS18].

Therefore, it is very surprising that the standard greedy algorithm does not only beat the 2-approximation on autonomous systems graphs like the internet [PW05], it also performs well on many real-world networks, obtaining

approximation ratios that are very close to 1 [SGS13]. So, again, there is a gap between the theoretical bounds and what is observed in practice, and we can get more realistic insights by focusing on networks featuring properties that are also observed in the real world, like a power-law degree distribution, high clustering, and the small-world phenomenon, as explained in Chapter 1.

Previous efforts to obtain better insights into the approximability of the vertex cover problem have focused on networks that feature only one of these properties, namely a power-law degree distribution [CFR16; GH14; VS16]. With this approach, guarantees for the approximation factor of the standard greedy algorithm were improved to a constant, compared to $\log(n)$ on general graphs [CFR16]. Moreover, it was shown that it is possible to compute an expected $(2 - \varepsilon)$-approximation for a constant ε, in polynomial time on such networks [GH14], and this was later improved to about 1.7 depending on properties of the distribution [VS16]. However, it was shown that even on graphs with a power-law degree distribution, the vertex cover problem remains NP-hard to approximate within some constant factor [CFR16]. This indicates that focusing on networks that only feature a power-law degree distribution, is not sufficient to explain why vertex cover can be approximated so well in practice.

The goal of this chapter is to close this gap between theory and practice, by considering the hyperbolic random graph model that features all of the three mentioned properties of scale-free networks, as explained in Section 3.3. To this end, we link the success of the standard greedy approach to structural properties of hyperbolic random graphs, identify the parts of the graph where it does not behave optimally, and use these insights to derive a new approximation algorithm. On the giant component of a hyperbolic random graph, this algorithm achieves an approximation ratio of $1 + o(1)$, asymptotically almost surely, and maintains an efficient running time of $O\big(m \log(n)\big)$. Since the average degree of hyperbolic random graphs is constant with high probability [Keu18], this implies a quasi-linear running time on such networks. Moreover, we introduce a parameter that can be used to tune the trade-off between approximation quality and running time of the algorithm, facilitating an improvement over the standard greedy approach. While our algorithm depends on the coordinates of the vertices in the hyperbolic plane, we propose an adaptation of it that is oblivious to the underlying geometry and compare its approximation performance to the standard greedy algorithm on a selection of real-world networks. On average our algorithm reduces the error of the standard greedy approach to less than 50%.

7.2 An Improved Greedy Algorithm

In the previous chapter, we considered the *dominance reduction rule*, which reduces a hyperbolic random graph to a remainder of simple structure, see Section 6.2.1. This rule states that a vertex u can be safely added to the vertex cover (and, thus, be removed from the graph) if it *dominates* at least one other vertex, i.e., if there exists a neighbor $v \in N(u)$ such that all neighbors of v are also neighbors of u.

On hyperbolic random graphs, vertices near the center of the disk dominate with high probability (Lemma 6.4). Therefore, it is not surprising that the standard greedy algorithm that computes a vertex cover by repeatedly taking the vertex with the largest degree achieves good approximation rates on such networks: Since high-degree vertices are near the disk center, the algorithm essentially favors vertices that are likely to dominate and can be safely added to the vertex cover anyway.

On the other hand, after (safely) removing high-degree vertices, the remaining vertices all have similar (small) degree, meaning the standard greedy algorithm basically picks the vertices at random. Thus, in order to improve the approximation performance of the algorithm, one has to improve on the parts of the graph that contain the low-degree vertices. Based on this insight, we derive a new greedy algorithm that achieves close to optimal approximation rates efficiently. More formally, we prove the following main theorem.

Theorem 7.1. *Let G be the giant component of a hyperbolic random graph. Given the radii of the vertices, an approximate vertex cover of G can be computed in time $\mathrm{O}\big(m \log(n)\big)$, such that the approximation ratio is $\big(1 + \mathrm{o}(1)\big)$ asymptotically almost surely.*

Consider the following greedy algorithm that computes an approximation of a minimum vertex cover on hyperbolic random graphs. We iterate the vertices in order of increasing radius. Each encountered vertex v is added to the cover and removed from the graph. After each step, we then identify the connected components of size at most $\tau \log\log(n)$ in the remainder of the graph, solve them optimally, and remove them from the graph as well. The constant $\tau > 0$ can be used to adjust the trade-off between quality and running time.

This algorithm determines the order in which the vertices are processed based on their radii, which are not known for real-world networks. However, in hyperbolic random graphs, there is a strong correlation between the radius of a

vertex and its degree, see Section 3.3.3. Therefore, we can mimic the considered greedy strategy by removing vertices with decreasing degree instead. Then, the above algorithm represents an adaptation of the standard greedy algorithm: Instead of greedily adding vertices with decreasing degree until all remaining vertices are isolated, we increase the quality of the approximation by solving small components exactly.

7.3 Approximation Performance

To analyze the performance of the above algorithm, we utilize structural properties of hyperbolic random graphs. While the power-law degree distribution and high clustering are modelled explicitly using the underlying geometry, other properties of the model, like the logarithmic diameter, emerge as a natural consequence of the first two. Our analysis is based on another emerging property: Hyperbolic random graphs decompose into small components when removing high-degree vertices.

More formally, we proceed as follows. We compute the size of the vertex cover obtained using the above algorithm, by partitioning the vertices of the graph into two sets: V_{Greedy} and V_{Exact}, denoting the vertices that were added greedily and the ones contained in small separated components that were solved exactly, respectively (see Figure 7.1). Clearly, we obtain a valid vertex cover for the whole graph, if we take all vertices in V_{Greedy} together with a vertex cover C_{Exact} of $G[V_{\text{Exact}}]$. Then, the approximation ratio is given by the quotient

$$\xi = \frac{|V_{\text{Greedy}}| + |C_{\text{Exact}}|}{|C_{\text{OPT}}|},$$

where C_{OPT} denotes an optimal solution. Since all components in $G[V_{\text{Exact}}]$ are solved optimally and since any minimum vertex cover for the whole graph induces a vertex cover on $G[V']$ for any vertex subset $V' \subseteq V$, it holds that $|C_{\text{Exact}}| \leq |C_{\text{OPT}}|$. Consequently, it suffices to show that $|V_{\text{Greedy}}| = o(|C_{\text{OPT}}|)$ in order to obtain the claimed approximation factor of $1 + o(1)$.

To bound the size of V_{Greedy}, we identify a time during the execution of the algorithm at which only few vertices were added greedily, yet, the majority of the vertices were contained in small separated components (and were, therefore, part of V_{Exact}), and only few vertices remain to be added greedily. Since the algorithm processes the vertices by increasing radius, this point in time can be translated to

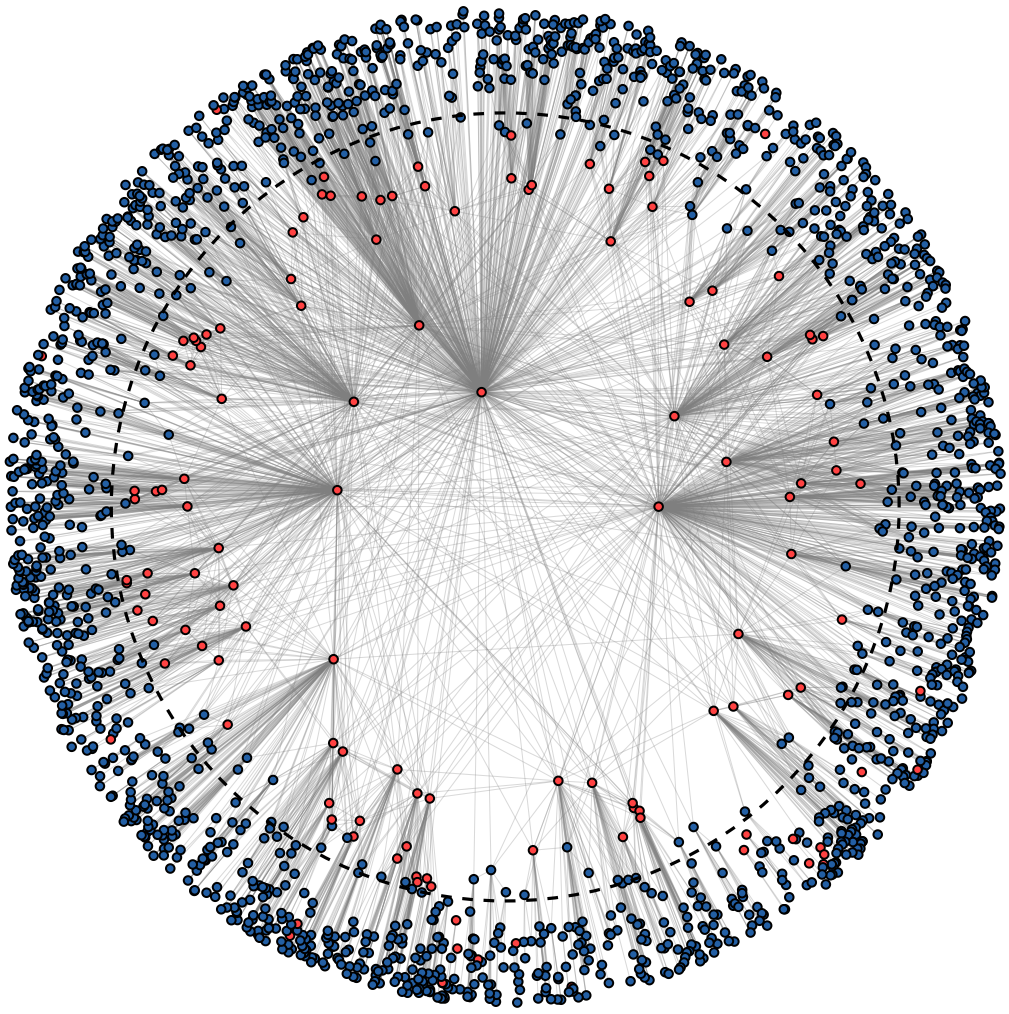


Figure 7.1: A hyperbolic random graph with 1942 vertices, average degree 7.7, and power-law exponent 2.6. The vertex sets V_{Greedy} and V_{Exact} are shown in red and blue, respectively. The dashed line shows a possible threshold radius ρ.

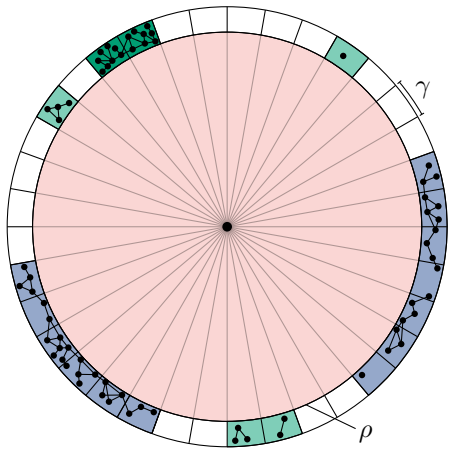


Figure 7.2: The disk is divided into the inner disk (red) and the outer band. It is additionally divided into sectors of equal width γ. Consecutive non-empty sectors form a run. Wide runs (blue) consist of many sectors. Each blue sector is a widening sector. Narrow runs (green) consist of few sectors. Small narrow runs contain only few vertices (light green), while large narrow runs contain many vertices (dark green).

a threshold radius ρ in the hyperbolic disk (see Figure 7.1). Therefore, we divide the disk into two regions: an *inner disk* and an *outer band*, containing vertices with radii below and above ρ, respectively. The threshold ρ is chosen such that a hyperbolic random graph decomposes into small components after removing the inner disk. When adding the first vertex from the outer band, greedily, we can assume that the inner disk is empty (since vertices of smaller radii were chosen before or removed as part of a small component). At this point, the majority of the vertices in the outer band were contained in small components, which have been solved exactly. In our analysis, we now overestimate the size of V_{Greedy} by assuming that all remaining vertices are also added to the cover greedily. Therefore, we obtain a valid upper bound on $|V_{\text{Greedy}}|$, by counting the total number of vertices in the inner disk and adding the number of vertices in the outer band that are contained in components that are not solved exactly, i.e., components whose size exceeds $\tau \log\log(n)$. In the following, we show that both numbers are sublinear in n with high probability. Together with the fact that an optimal vertex cover on hyperbolic random graphs, asymptotically almost surely, contains $\Omega(n)$ vertices [CFR16], this implies $|V_{\text{Greedy}}| = o(|C_{\text{OPT}}|)$.

The main contribution of our analysis is the identification of small components in the outer band, which is done by discretizing it into sectors, such that an edge cannot extend beyond an empty sector (see Figure 7.2). The foundation of this analysis is the delicate interplay between the angular width γ of these sectors and the threshold ρ that defines the outer band. Recall that ρ is used to represent the time in the execution of the algorithm at which the graph has been decomposed into small components. For our analysis we assume that all vertices seen before this point (all vertices in the inner disk; red in Figure 7.2) were added greedily. Therefore, if we choose ρ too large, we overestimate the actual number of greedily added vertices by too much. As a consequence, we want to choose ρ as small as possible. However, this conflicts our intentions for the choice of γ and its impact on ρ. Recall that the maximum angular distance between two vertices such that they are adjacent increases with decreasing radii (Section 3.3.5). Thus, in order to avoid edges that extend beyond an angular width of γ, we need to ensure that the radii of the vertices in the outer band are sufficiently large. That is, decreasing γ requires increasing ρ. However, we want to make γ as small as possible, in order to get a finer granularity in the discretization and, with that, a more accurate analysis of the component structure in the outer band. Therefore, γ and ρ need to be chosen such that the inner disk does not become too large, while ensuring that the discretization is granular enough to accurately detect components whose size depends on τ and n. To this end, we adjust the angular width of the sectors using a function $\gamma(n, \tau)$, which is defined as

$$\gamma(n, \tau) = \log\left(\frac{\tau \log^{(2)}(n)}{2 \log^{(3)}(n)^2}\right),$$

where $\log^{(i)}(n)$ denotes iteratively applying the log-function i times on n (e.g., $\log^{(2)}(n) = \log\log(n)$), and set

$$\rho = R - \log(\pi/2 \cdot e^{C/2}\gamma(n, \tau)),$$

where $R = 2\log(n) + C$ is the radius of the hyperbolic disk (see Section 3.3).

In the following, we first show that the number of vertices in the inner disk is sublinear with high probability, before analyzing the component structure in the outer band. To this end, we make use of the discretization of the disk into sectors, by distinguishing between different kinds of *runs* (sequences of

non-empty sectors), see Figure 7.2. In particular, we bound the number of *wide* runs (consisting of many sectors) and the number of vertices in them. Then we bound the number of vertices in *large narrow* runs (consisting of few sectors but containing many vertices). The remaining *small narrow* runs represent small components that are solved exactly.

7.3.1 The Inner Disk

The inner disk $\mathcal{D}_\rho$ contains all vertices whose radius is below the threshold ρ. The number of them that are added to the cover greedily is bounded by the number of all vertices in $\mathcal{D}_\rho$.

Lemma 7.2. *Let G be a hyperbolic random graph on n vertices with power-law exponent $\beta = 2\alpha + 1$. Then, with high probability, $\left|V(\mathcal{D}_\rho)\right| = \mathrm{O}\big(n \cdot \gamma(n,\tau)^{-\alpha}\big)$.*

Proof. We start by computing the expected number of vertices in $\mathcal{D}_\rho$ and show concentration afterwards. To this end, we first compute the measure $\mu(\mathcal{D}_\rho)$. The measure of a disk of radius r that is centered at the origin is given by $e^{-\alpha(R-r)}(1+\mathrm{o}(1))$, see Equation (3.8). Consequently, the expected number of vertices in $\mathcal{D}_\rho$ is

$$\begin{aligned}\mathbb{E}\Big[\left|V(\mathcal{D}_\rho)\right|\Big] &= n\mu\Big(\mathcal{D}_\rho\Big)\\ &= \mathrm{O}\Big(ne^{-\alpha(R-\rho)}\Big)\\ &= \mathrm{O}\Big(ne^{-\alpha\log(\pi/2\cdot e^{C/2}\gamma(n,\tau))}\Big)\\ &= \mathrm{O}\big(n\cdot\gamma(n,\tau)^{-\alpha}\big).\end{aligned}$$

Since $\gamma(n,\tau) = \mathrm{O}\big(\log^{(3)}(n)\big)$, this bound on $\mathbb{E}\big[|V(\mathcal{D}_\rho)|\big]$ is $\omega\big(\log(n)\big)$, and we can apply the Chernoff bound in Corollary 2.3 to conclude that $\left|V(\mathcal{D}_\rho)\right| = \mathrm{O}\big(n\cdot\gamma(n,\tau)^{-\alpha}\big)$ holds with probability $1-\mathrm{O}\big(n^{-c}\big)$ for any $c>0$. □

Since $\gamma(n,\tau) = \omega(1)$, Lemma 7.2 shows that, with high probability, the number of vertices that are greedily added to the vertex cover in the inner disk is sublinear. Once the inner disk has been processed and removed, the graph has been decomposed into small components and the ones of size at most $\tau\log\log(n)$ have already been solved exactly. The remaining vertices that are now added greedily belong to large components in the outer band.

7.3.2 The Outer Band

To identify the vertices in the outer band that are contained in components whose size exceeds $\tau \log\log(n)$, we divide it into sectors of angular width

$$\gamma = \theta(\rho, \rho) = \pi \cdot \frac{\gamma(n, \tau)}{n} \cdot \big(1 + o(1)\big),$$

where $\theta(\rho, \rho)$ denotes the maximum angular distance between two vertices with radii ρ to be adjacent (see Section 3.3.5). This division is depicted in Figure 7.2. The choice of γ (combined with the choice of ρ) has the effect that an edge between two vertices in the outer band cannot extend beyond an empty sector, i.e., a sector that does not contain any vertices, allowing us to use empty sectors as delimiters between components. To this end, we introduce the notion of *runs*, which are maximal sequences of non-empty sectors (see Figure 7.2). While a run can contain multiple components, the number of vertices in it denotes an upper bound on the combined sizes of the components that it contains.

To show that there are only few vertices in components whose size exceeds $\tau \log\log(n)$, we bound the number of vertices in runs containing more than this many vertices. For a given run this can happen for two reasons. First, it may contain many vertices if its angular interval is too large, i.e., it consists of too many sectors. This is unlikely, since the sectors are chosen sufficiently small, such that the probability for a given one to be empty is high. Second, while the angular width of the run is not too large, it contains too many vertices for its size. However, the vertices of the graph are distributed uniformly at random in the disk, making it unlikely that too many vertices are sampled into such a small area. To formalize this, we introduce a threshold w and distinguish between two types of runs: A *wide* run contains more than w sectors, while a *narrow* run contains at most w sectors. The threshold w is chosen such that the probabilities for a run to be wide and for a narrow run to contain more than $\tau \log\log(n)$ vertices are small. To this end, we set $w = e^{\gamma(n,\tau)} \cdot \log^{(3)}(n)$.

In the following, we first bound the number of vertices in wide runs. Afterwards, we consider narrow runs that contain more than $\tau \log\log(n)$ vertices. Together, this gives an upper bound on the number of vertices that are added greedily in the outer band.

Wide Runs

We refer to a sector that contributes to a wide run as a *widening sector*. In the following, we bound the number of vertices in all wide runs in three steps. First, we determine the expected number of all widening sectors. Second, based on the expected value, we show that the number of widening sectors is small, with high probability. Finally, we make use of the fact that the area of the disk covered by widening sectors is small, to show that the number of vertices sampled into the corresponding area is sublinear, with high probability.

Expected Number of Widening Sectors. Let n' denote the total number of sectors and let $\mathcal{S}_1, \dots, \mathcal{S}_{n'}$ be the corresponding sequence. For each sector $\mathcal{S}_k$, we define the random variable S_k indicating whether $\mathcal{S}_k$ contains any vertices, i.e., $S_k = 0$ if $V(\mathcal{S}_k) = \emptyset$ and $S_k = 1$ otherwise. The sectors in the disk are then represented by a circular sequence of indicator random variables $S_1, \dots, S_{n'}$, and we are interested in the random variable W that denotes the sum of all runs of 1s that are longer than w. In order to compute $\mathbb{E}[W]$, we first compute the total number of sectors, as well as the probability for a sector to be empty or non-empty.

Lemma 7.3. *Let G be a hyperbolic random graph on n vertices. Then, the number of sectors of width $\gamma = \theta(\rho, \rho)$ is $n' = 2n/\gamma(n,\tau) \cdot \big(1 \pm \mathrm{o}(1)\big)$.*

Proof. Since all sectors have equal angular width $\gamma = \theta(\rho, \rho)$, we can use Equation (3.10) to compute the total number of sectors as

$$n' = \frac{2\pi}{\theta(\rho,\rho)} = \pi e^{-R/2+\rho}\left(1 \pm \Theta\left(e^{R-2\rho}\right)\right)^{-1}.$$

By substituting $\rho = R - \log(\pi/2 \cdot e^{C/2}\gamma(n,\tau))$ and $R = 2\log(n) + C$, we obtain

$$n' = \frac{\pi e^{R/2}}{\pi/2 \cdot e^{C/2}\gamma(n,\tau)}\left(1 \pm \Theta\left(e^{-R}\gamma(n,\tau)^2\right)\right)^{-1} = \frac{2n}{\gamma(n,\tau)}\left(1 \pm \Theta\left(\left(\frac{\gamma(n,\tau)}{n}\right)^2\right)\right)^{-1}.$$

It remains to simplify the error term. Note that $\gamma(n,\tau) = \mathrm{O}\big(\log^{(3)}(n)\big)$. Consequently, the error term is equivalent to $\big(1 \pm \mathrm{o}(1)\big)^{-1}$. Finally, it holds that $1/(1+x) = 1 - \Theta(x)$ for $x = \pm\mathrm{o}(1)$, according to Lemma 2.11. □

Lemma 7.4. *Let G be a hyperbolic random graph on n vertices and let $\mathcal{S}$ be a sector of angular width $\gamma = \theta(\rho, \rho)$. For sufficiently large n, the probability that $\mathcal{S}$ contains at least one vertex is bounded by*

$$1 - e^{-\gamma(n,\tau)/4} \leq \Pr\big[V(\mathcal{S}) \neq \emptyset\big] \leq e^{-\left(e^{-\gamma(n,\tau)}\right)}.$$

Proof. To compute the probability that $\mathcal{S}$ contains at least one vertex, we first compute the probability for a given vertex to be sampled into $\mathcal{S}$, which is given by the measure $\mu(\mathcal{S})$. Since the angular coordinates of the vertices are distributed uniformly at random and since the disk is divided into n' sectors of equal width, the measure of a single sector $\mathcal{S}$ can be obtained as $\mu(\mathcal{S}) = 1/n'$. The total number of sectors n' is given by Lemma 7.3 and we can derive

$$\mu(\mathcal{S}) = \frac{\gamma(n,\tau)}{2n}\big(1 \pm \mathrm{o}(1)\big)^{-1} = \frac{\gamma(n,\tau)}{2n}\big(1 \pm \mathrm{o}(1)\big),$$

where the second equality is obtained by applying $1/(1+x) = 1 - \Theta(x)$ for $x = \pm\mathrm{o}(1)$, see Lemma 2.11.

Given $\mu(\mathcal{S})$, we first compute the lower bound on the probability that $\mathcal{S}$ contains at least one vertex. Note that

$$\Pr\big[V(\mathcal{S}) \neq \emptyset\big] = 1 - \Pr\big[V(\mathcal{S}) = \emptyset\big]. \tag{7.1}$$

Therefore, it suffices to show that $\Pr\big[V(\mathcal{S}) = \emptyset\big] \leq e^{-\gamma(n,\tau)/4}$. The probability that $\mathcal{S}$ is empty is $(1 - \mu(\mathcal{S}))^n$. By Lemma 2.9 it holds that $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$. Consequently, we have

$$\Pr\big[V(\mathcal{S}) = \emptyset\big] \leq e^{-n\mu(\mathcal{S})} \leq e^{-\gamma(n,\tau)/2 \cdot \left(1-\mathrm{o}(1)\right)}$$

and for large enough n it holds that $1 - \mathrm{o}(1) \geq 1/2$.

It remains to compute the upper bound. Again, using Equation (7.1) and since $\Pr\big[V(\mathcal{S}) = \emptyset\big] = (1 - \mu(\mathcal{S}))^n$, we can compute the probability that $\mathcal{S}$ contains at least one vertex as

$$\Pr\big[V(\mathcal{S}) \neq \emptyset\big] = 1 - \big(1 - \mu(\mathcal{S})\big)^n.$$

Note that $\mu(\mathcal{S}) = \mathrm{o}(1)$. Therefore, we can apply Lemma 2.10, which states that

$1 - x \geq e^{-x(1+o(1))}$ for $x = o(1)$, and obtain the following upper bound

$$\begin{aligned}\Pr\big[V(\mathcal{S}) \neq \emptyset\big] &= 1 - \big(1 - \mu(\mathcal{S})\big)^n \\ &\leq 1 - e^{-n\mu(\mathcal{S})\left(1+o(1)\right)} \\ &\leq 1 - e^{-\gamma(n,\tau)/2\cdot\left(1+o(1)\right)}.\end{aligned}$$

For large enough n, we have $\big(1 + o(1)\big) \leq 2$. Therefore,

$$\Pr\big[V(\mathcal{S}) \neq \emptyset\big] \leq 1 - e^{-\gamma(n,\tau)}$$

holds for sufficiently large n. Finally, $1 - x \leq e^{-x}$ is valid for all $x \in \mathbb{R}$ by Lemma 2.9, and we obtain the claimed bound. □

We are now ready to compute an upper bound on the expected number of widening sectors, i.e., sectors that are part of wide runs. To this end, we aim to apply the following lemma.

Lemma 7.5 ([MPP07, Proposition 4.3[6]]). *Let $S_1, \dots, S_{n'}$ denote a circular sequence of independent indicator random variables, such that $\Pr[S_k = 1] = p$ and $\Pr[S_k = 0] = 1 - p = q$, for all $k \in [n']$. Furthermore, let W denote the sum of the lengths of all success runs of length at least $w \leq n'$. Then, $\mathbb{E}[W] = n' p^w (wq + p)$.*

We note that the indicator random variables $S_1, \dots, S'_n$ are *not* independent on hyperbolic random graphs. To overcome this issue, we compute the expected value of W on hyperbolic random graphs with n vertices in expectation (see Section 3.3.4) and subsequently derive a probabilistic bound on W for hyperbolic random graphs.

Lemma 7.6. *Let G be a hyperbolic random graph with n vertices in expectation and let W denote the number of widening sectors. Then,*

$$\mathbb{E}[W] \leq \frac{2^{1/4} \cdot \tau^{3/4} \cdot n}{\gamma(n,\tau) \cdot \log^{(2)}(n)^{1/4} \cdot \log^{(3)}(n)^{1/2}} \big(1 \pm o(1)\big).$$

6 The original statement has been adapted to fit our notation. We use n', w, and W to denote the total number of random variables, the threshold for long runs, and the sum of their lengths, respectively. They were previously denoted by n, k, and S, respectively. In the original statement $s = 0$ indicates that the variables are distributed independently and identically, and c indicates that the sequence is circular.

Proof. A widening sector is part of a run of more than $w = e^{\gamma(n,\tau)} \cdot \log^{(3)}(n)$ consecutive non-empty sectors. To compute the expected number of widening sectors, we apply Lemma 7.5. To this end, we use Lemma 7.3 to bound the total number of sectors n' and Lemma 7.4 to bound the probability $p = \Pr[S_k = 1]$ (i.e., the probability that sector $\mathcal{S}_k$ is not empty) as $p \leq \exp(-(e^{-\gamma(n,\tau)}))$, as well as the complementary probability $q = 1 - p \leq e^{-\gamma(n,\tau)/4}$. We obtain

$$\begin{aligned}
\mathbb{E}[W] &= n' p^{(w+1)}((w+1)q + p) \\
&\leq \frac{2n}{\gamma(n,\tau)}\big(1 \pm \mathrm{o}(1)\big) \cdot e^{-\left((w+1)e^{-\gamma(n,\tau)}\right)} \cdot \left((w+1)e^{-\frac{\gamma(n,\tau)}{4}} + 1\right) \\
&\leq \frac{2n}{\gamma(n,\tau)} e^{\left(-e^{\gamma(n,\tau)} \log^{(3)}(n) e^{-\gamma(n,\tau)}\right)} \\
&\qquad \cdot \left(\left(e^{\gamma(n,\tau)} \log^{(3)}(n) + 1\right)e^{-\frac{\gamma(n,\tau)}{4}} + 1\right)\big(1 \pm \mathrm{o}(1)\big).
\end{aligned}$$

Now the first exponential simplifies to $\exp(-\log^{(3)}(n)) = \log^{(2)}(n)^{-1}$, since the $\gamma(n,\tau)$ terms cancel. Factoring out $\exp(3/4 \cdot \gamma(n,\tau)) \log^{(3)}(n)$ in the third term then yields

$$\begin{aligned}
\mathbb{E}[W] \leq{}& \frac{2ne^{3/4 \cdot \gamma(n,\tau)} \log^{(3)}(n)}{\gamma(n,\tau) \cdot \log^{(2)}(n)} \\
&\cdot \left(1 + \frac{1}{e^{\gamma(n,\tau)} \log^{(3)}(n)} + \frac{1}{e^{3/4 \cdot \gamma(n,\tau)} \log^{(3)}(n)}\right)\big(1 \pm \mathrm{o}(1)\big).
\end{aligned}$$

Since $\gamma(n,\tau) = \omega(1)$, the first error term can be simplified as $\big(1 + \mathrm{o}(1)\big)$. Additionally, we can substitute $\gamma(n,\tau) = \log(\tau \log^{(2)}(n)/(2 \log^{(3)}(n)^2))$ to obtain

$$\mathbb{E}[W] \leq 2^{1/4} \frac{\tau^{3/4} \cdot n \cdot \log^{(3)}(n)}{\gamma(n,\tau) \cdot \log^{(2)}(n)} \cdot \frac{\log^{(2)}(n)^{3/4}}{\log^{(3)}(n)^{3/2}} \cdot \big(1 \pm \mathrm{o}(1)\big).$$

Further simplification then yields the claim. □

Concentration for the Number of Widening Sectors. The above lemma bounds the expected number of widening sectors and it remains to show that this bound holds with high probability. To this end, we first determine under which conditions the sum of long success runs in a circular sequence of indicator

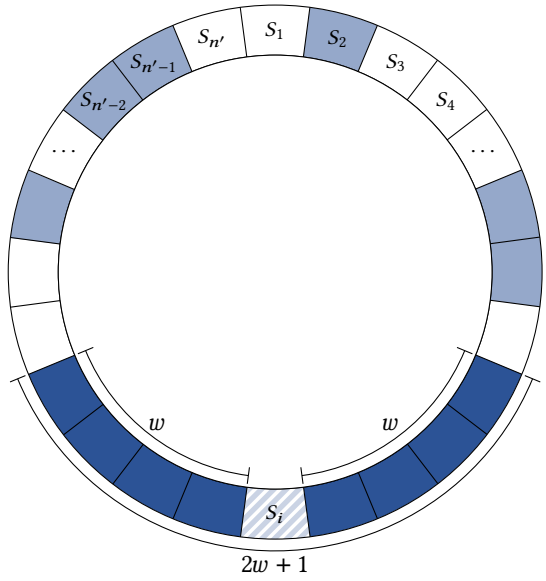


Figure 7.3: A circular sequence of random variables $S_1, \dots, S_{n'}$ that can either be 0 (white) or 1 (blue). Dark blue runs are as large as possible without being wide. Depending on the value of S_i, the two runs of length w are merged into one run of length $2w + 1$.

random variables can be bounded with high probability in general. Afterwards, we show that these conditions are met for our application.

Lemma 7.7. *Let $S_1, \dots, S_{n'}$ denote a circular sequence of independent indicator random variables and let W denote the sum of the lengths of all success runs of length at least $1 \leq w \leq n'$. If $g(n') = \omega\big(w\sqrt{n' \log(n')}\big)$ is an upper bound on $\mathbb{E}[W]$, then $W = \mathrm{O}\big(g(n')\big)$ holds with probability $1 - \mathrm{O}\big((n')^{-c}\big)$ for any constant c.*

Proof. In order to show that W does not exceed $g(n')$ by more than a constant factor with high probability, we aim to apply a method of bounded differences (Corollary 2.6). To this end, we consider W as a function of n' independent random variables $S_1, \dots, S_{n'}$ and determine the parameters Δ_i with which W satisfies the bounded differences condition (see Equation (2.2)). That is, for each $i \in [n']$ we need to bound the change in the sum of the lengths of all success runs of length at least w, obtained by changing the value of S_i from 0 to 1 or vice versa.

The largest impact on W is obtained when changing the value of S_i from 0 to 1 merges two runs of size w, i.e., runs that are as large as possible but not *wide*, as shown in Figure 7.3. In this case both runs did not contribute anything to W before the change, while the merged run now contributes $2w + 1$. Then, we can bound the change in W as $\Delta_i = 2w + 1$. Note that the other case in which the value of S_i is changed from 1 to 0 can be viewed as the inversion of the change in the first case. That is, instead of merging two runs, changing S_i splits a single run into two. Consequently, the corresponding bound on the change of W is the same, except that W is decreasing instead of increasing.

Thus, W satisfies the bounded differences condition for $\Delta_i = 2w + 1$ for all $i \in [n']$. By Corollary 2.6, we can now bound the probability that W exceeds an upper bound $g(n')$ on its expected value by more than a constant factor as

$$\Pr\left[W > c_1 g(n')\right] \leq e^{-2((c_1-1)g(n'))^2/\Delta},$$

where $\Delta = \sum_{i\in[n']} \Delta_i^2$ and $c_1 \geq 1$. Since we have $\Delta_i = 2w + 1$ for all $i \in [n']$, it follows that $\Delta = n'(2w + 1)^2$. Thus,

$$\begin{aligned}\Pr\left[W > c_1 g(n')\right] &\leq e^{-\frac{2((c_1-1)g(n'))^2}{n'(2w+1)^2}} \\ &\leq e^{-\frac{2(c_1-1)^2}{n'}\cdot\left(\frac{g(n')}{3w}\right)^2},\end{aligned}$$

where the second inequality is valid since w is assumed to be at least 1. Moreover, we can apply $g(n') = \omega\left(w\sqrt{n' \log(n')}\right)$ (a precondition of this lemma), which yields

$$\begin{aligned}\Pr\left[W > c_1 g(n')\right] &\leq e^{-\frac{2(c_1-1)^2}{9n'}\cdot\left(\omega\left(\sqrt{n' \log(n')}\right)\right)^2} \\ &= e^{-\omega\left(\log(n')\right)} \\ &= (n')^{-\omega(1)}.\end{aligned}$$

As a consequence, for any constant c, it holds that

$$\begin{aligned}\Pr\left[W = O\left(g(n')\right)\right] &= 1 - (n')^{-\omega(1)} \\ &= 1 - O\left((n')^{-c}\right).\end{aligned}$$

□

Lemma 7.8. *Let G be a hyperbolic random graph on n vertices. Then, with probability $1 - O(n^{-c})$ for any constant $c > 0$, the number of widening sectors W is bounded by*

$$W = O\left(\frac{\tau^{3/4} \cdot n}{\gamma(n,\tau) \cdot \log^{(2)}(n)^{1/4} \cdot \log^{(3)}(n)^{1/2}}\right).$$

Proof. We show that the claimed bound holds with probability $1 - O(n^{-c_1})$ for any constant $c_1 > 0$ on hyperbolic random graphs with n vertices in expectation. By Lemma 3.9 the same bound then holds with probability $1 - O(n^{-c_1+1/2})$ on hyperbolic random graphs. Choosing $c = c_1 - 1/2$ then yields the claim. Recall that we represent the sectors using a circular sequence of independent indicator random variables $S_1, \dots, S_{n'}$ and that W denotes the sum of the lengths of all success runs spanning more than w sectors, i.e., the sum of all widening sectors. By Lemma 7.6 we obtain an upper bound on $\mathbb{E}[W]$ by choosing

$$g(n') = h(n) = \frac{2^{1/4} \cdot \tau^{3/4} \cdot n}{\gamma(n,\tau) \cdot \log^{(2)}(n)^{1/4} \cdot \log^{(3)}(n)^{1/2}}\big(1 \pm o(1)\big)$$

and it remains to show that this bound holds with sufficiently high probability. To this end, we aim to apply Lemma 7.7, which states that $W = O(g(n'))$ holds with probability $1 - O((n')^{-c_2})$ for any constant c_2, if $g(n') = \omega\big(w\sqrt{n' \log(n')}\big)$. In the following, we first show that $h(n)$ fulfills this criterion[7], before arguing that we can choose c_2 such that $1 - O((n')^{-c_2}) = 1 - O(n^{-c_1})$ for any constant c_1. Since $\tau = \Theta(1)$ and $n' = \Theta(n/\gamma(n,\tau))$ by Lemma 7.3, we can bound $h(n)$ by

$$\begin{aligned} h(n) &= \Theta\left(\frac{n'}{\log^{(2)}(n)^{1/4} \log^{(3)}(n)^{1/2}}\right) \\ &= \Theta\left(\frac{\log^{(2)}(n) \cdot n'}{\log^{(2)}(n)^{5/4} \log^{(3)}(n)^{1/2}}\right) \\ &= \omega\left(\frac{\log^{(2)}(n)}{\log^{(3)}(n)} \cdot \frac{n'}{\log^{(2)}(n)^{5/4}}\right), \end{aligned}$$

7 Note that since we are interested in runs of *strictly more than* w sectors, we need to show $g(n') = \omega((w+1)\sqrt{n' \log(n')})$. However, it is easy to see that this is implied by showing $g(n') = \omega(w\sqrt{n' \log(n')})$.

where the last bound is obtained by applying $\log^{(3)}(n)^{1/2} = \omega(1)$. Recall that w was chosen as $w = e^{\gamma(n,\tau)} \log^{(3)}(n)$. Furthermore, we have

$$\gamma(n,\tau) = \log\left(\tau \log^{(2)}(n) / \left(2\log^{(3)}(n)^2\right)\right).$$

Thus, it holds that $w = \Theta\left(\log^{(2)}(n)/(\log^{(3)}(n))\right)$, allowing us to further bound $h(n)$ by

$$\begin{aligned} h(n) &= \omega\left(w \frac{n'}{\log^{(2)}(n)^{5/4}}\right) \\ &= \omega\left(w \sqrt{n' \cdot \frac{n'}{\log^{(2)}(n)^{5/2}}}\right) \\ &= \omega\left(w \sqrt{n' \log(n') \cdot \frac{n'}{\log(n') \log^{(2)}(n)^{5/2}}}\right). \end{aligned}$$

To finish the proof, it now remains to show that the last factor in the root is $\omega(1)$. Note that $n' = \Omega\left(n/\log^{(3)}(n)\right)$ and $n' = O(n)$. Consequently, it holds that

$$\begin{aligned} \frac{n'}{\log(n') \log^{(2)}(n)^{5/2}} &= \Omega\left(\frac{n}{\log(n) \cdot \log^{(2)}(n)^{5/2} \cdot \log^{(3)}(n)}\right) \\ &= \omega\left(\frac{n}{\log(n)^3}\right) \\ &= \omega(1). \end{aligned}$$

As stated above, we thus have $W = O\left(h(n)\right)$ with probability $1 - O\left((n')^{-c_2}\right)$ for any constant c_2. Again, since $n' = \Omega\left(n/\log^{(3)}(n)\right)$, we have $n' = \Omega\left(n^{1/2}\right)$. Therefore, we can conclude that $W = O\left(h(n)\right)$ holds with probability $1 - O\left(n^{-c_2/2}\right)$. Choosing $c_2 = 2c_1$ then yields the claim. □

Number of Vertices in Wide Runs. Let $\mathcal{W}$ denote the area of the disk covered by all widening sectors. By Lemma 7.8 the total number of widening sectors is small, with high probability. As a consequence, $\mathcal{W}$ is small as well and we can derive that the size of the vertex set $V(\mathcal{W})$ containing all vertices in all widening sectors is sublinear with high probability.

Lemma 7.9. *Let G be a hyperbolic random graph on n vertices. Then, with high probability, the number of vertices in wide runs is bounded by*

$$\left|V(\mathcal{W})\right| = \mathrm{O}\left(\frac{\tau^{3/4}\cdot n}{\log^{(2)}(n)^{1/4}\cdot\log^{(3)}(n)^{1/2}}\right).$$

Proof. We start by computing the expected number of vertices in $\mathcal{W}$ and show concentration afterwards. The probability for a given vertex to fall into $\mathcal{W}$ is equal to its measure $\mu(\mathcal{W})$. Since the angular coordinates of the vertices are distributed uniformly at random, we have $\mu(\mathcal{W}) = W/n'$, where W denotes the number of widening sectors and n' is the total number of sectors, which is given by Lemma 7.3. The expected number of vertices in $\mathcal{W}$ is then

$$\mathbb{E}\Big[\left|V(\mathcal{W})\right|\Big] = n\mu(\mathcal{W}) = n\frac{W}{n'} = \frac{1}{2}W\cdot\gamma(n,\tau)\big(1\pm\mathrm{o}(1)\big), \tag{7.2}$$

where the last equality holds since $1/(1+x) = 1-\Theta(x)$ is valid for $x = \pm\mathrm{o}(1)$, see Lemma 2.11. Note that the number of widening sectors W is itself a random variable. Therefore, we apply the law of total expectation and consider different outcomes of W weighted with their probabilities. Motivated by the previously determined probabilistic bound on W (Lemma 7.8), we consider the events $W \le g(n)$ and $W > g(n)$, where

$$g(n) = \frac{c\cdot\tau^{3/4}\cdot n}{\gamma(n,\tau)\cdot\log^{(2)}(n)^{1/4}\cdot\log^{(3)}(n)^{1/2}},$$

for sufficiently large $c > 0$ and n. With this, we can compute the expected number of vertices in $\mathcal{W}$ as

$$\begin{aligned}\mathbb{E}\Big[\left|V(\mathcal{W})\right|\Big] = \mathbb{E}\Big[\left|V(\mathcal{W})\right| \;\Big|\; W\le g(n)\Big]\cdot\Pr\big[W\le g(n)\big] + \\ \mathbb{E}\Big[\left|V(\mathcal{W})\right| \;\Big|\; W> g(n)\Big]\cdot\Pr\big[W> g(n)\big].\end{aligned}$$

To bound the first summand, note that $\Pr\big[W\le g(n)\big]\le 1$. Further, by applying Equation (7.2) from above, we have

$$\mathbb{E}\Big[\left|V(\mathcal{W})\right| \;\Big|\; W\le g(n)\Big]\cdot\Pr\big[W\le g(n)\big] \le \frac{1}{2}g(n)\cdot\gamma(n,\tau)\big(1\pm\mathrm{o}(1)\big).$$

In order to bound the second summand, note that n is an obvious upper bound on $\mathbb{E}\big[|V(\mathcal{W})|\big]$. Moreover, by Lemma 7.8 it holds that $\Pr\big[W > g(n)\big] = \mathrm{O}\big(n^{-c_1}\big)$ for any $c_1 > 0$. As a result we have

$$\mathbb{E}\Big[\big|V(\mathcal{W})\big| \,\Big|\, W > g(n)\Big] \cdot \Pr\big[W > g(n)\big] \leq n \cdot \Pr\big[W > g(n)\big] = \mathrm{O}\Big(n^{-c_1+1}\Big),$$

for any $c_1 > 0$. Clearly, the first summand dominates the second and we can conclude that $\mathbb{E}\big[|V(\mathcal{W})|\big] = \mathrm{O}\big(g(n)\gamma(n,\tau)\big)$. Consequently, for large enough n, there exists a constant $c_2 > 0$ such that $\hat{g}(n) = c_2 g(n)\gamma(n,\tau)$ is a valid upper bound on $\mathbb{E}\big[|V(\mathcal{W})|\big]$. This allows us to apply the Chernoff bound in Corollary 2.2 to bound the probability that $|V(\mathcal{W})|$ exceeds $\hat{g}(n)$ by more than a constant factor as

$$\Pr\Big[\big|V(\mathcal{W})\big| \geq (1+\varepsilon)\hat{g}(n)\Big] \leq e^{-\varepsilon^2/3 \cdot \hat{g}(n)}.$$

Finally, since $\hat{g}(n)$ can be simplified as

$$\hat{g}(n) = c_2 \cdot \frac{c \cdot \tau^{3/4} \cdot n}{\log^{(2)}(n)^{1/4} \cdot \log^{(3)}(n)^{1/2}},$$

it is easy to see that $\hat{g}(n) = \omega\big(\log(n)\big)$ and, therefore, $|V(\mathcal{W})| = \mathrm{O}\big(\hat{g}(n)\big)$ holds with probability $1 - \mathrm{O}\big(n^{-c_3}\big)$ for any $c_3 > 0$. □

Narrow Runs

It remains to bound the number of vertices in large components contained in narrow runs. In the following, we differentiate between *small* and *large* narrow runs, containing at most and more than $\tau \log\log(n)$ vertices, respectively. To obtain an upper bound on the number N of vertices in all large narrow runs, we determine the area $\mathcal{N}$ of the disk that is covered by them. We start by computing the expected number of vertices contained in a single narrow run from which we can derive that the probability for a narrow run to be large is low.

Expected Number of Vertices in Large Narrow Runs. The following lemma bounds the expected number of vertices in a given narrow run $\mathcal{R}$.

Lemma 7.10. *Let G be a hyperbolic random graph on n vertices and let $\mathcal{R}$ be a narrow run. Then, $\mathbb{E}\big[|V(\mathcal{R})|\big] \leq 1/2 \cdot e^{\gamma(n,\tau)} \log^{(3)}(n)\gamma(n,\tau)\big(1 \pm \mathrm{o}(1)\big)$.*

Proof. A narrow run consists of at most $w = e^{\gamma(n,\tau)} \log^{(3)}(n)$ sectors. Since the angular coordinates of the vertices are distributed uniformly at random and since we partitioned the disk into n' disjoint sectors of equal width, we can derive an upper bound on the expected number of vertices in $\mathcal{R}$ as $\mathbb{E}\big[|V(\mathcal{R})|\big] \leq nw/n'$. As $n' = 2n/\gamma(n,\tau) \cdot \big(1 \pm \mathrm{o}(1)\big)$ according to Lemma 7.3, we have

$$\mathbb{E}\Big[\big|V(\mathcal{R})\big|\Big] \leq 1/2 \cdot e^{\gamma(n,\tau)} \log^{(3)}(n)\gamma(n,\tau)\big(1 \pm \mathrm{o}(1)\big)^{-1}.$$

Since $1/(1+x) = 1 - \Theta(x)$ for $x = \pm\mathrm{o}(1)$, due to Lemma 2.11, we obtain the claimed bound. □

Using this upper bound, we can bound the probability that the number of vertices in a narrow run exceeds the threshold $\tau \log\log(n)$ by a certain amount.

Lemma 7.11. *Let G be a hyperbolic random graph on n vertices and let $\mathcal{R}$ be a narrow run. For $k > \tau \log\log(n)$ and n large enough, it holds that*

$$\Pr\Big[\big|V(\mathcal{R})\big| = k\Big] \leq e^{-k/18}.$$

Proof. First note that $\Pr\big[|V(\mathcal{R})| = k\big] \leq \Pr\big[|V(\mathcal{R})| \geq k\big]$. To show that this probability is small, we apply the Chernoff bound in Corollary 2.2, stating that

$$\Pr\Big[\big|V(\mathcal{R})\big| \geq (1+\varepsilon)g(n)\Big] \leq e^{-\varepsilon^2/3 \cdot g(n)}$$

holds for any $\varepsilon \in (0,1)$, if $g(n)$ is an upper bound on $\mathbb{E}\big[|V(\mathcal{R})|\big]$. To this end, we set $g(n) = 1/(1+\varepsilon) \cdot k$. To show that this is a valid choice, we can use Lemma 7.10 and substitute $\gamma(n,\tau) = \log(\tau \log^{(2)}(n)/(2\log^{(3)}(n)^2))$, which yields

$$\begin{aligned}
\mathbb{E}\Big[\big|V(\mathcal{R})\big|\Big] &\leq \frac{1}{2} e^{\gamma(n,\tau)} \log^{(3)}(n)\gamma(n,\tau)\big(1 \pm \mathrm{o}(1)\big) \\
&= \frac{\tau \log^{(2)}(n)}{4\log^{(3)}(n)^2} \cdot \log^{(3)}(n) \cdot \log\left(\frac{\tau \log^{(2)}(n)}{2\log^{(3)}(n)^2}\right)\big(1 \pm \mathrm{o}(1)\big) \\
&= \frac{\tau \log^{(2)}(n)}{4\log^{(3)}(n)} \cdot \Big(\log^{(3)}(n) - \big(2\log^{(4)}(n) - \log(\tau/2)\big)\Big)\big(1 \pm \mathrm{o}(1)\big) \\
&= \frac{1}{4} \cdot \tau \log^{(2)}(n) \cdot \left(1 - \frac{2\log^{(4)}(n) - \log(\tau/2)}{\log^{(3)}(n)}\right)\big(1 \pm \mathrm{o}(1)\big).
\end{aligned}$$

Note, that the first error term is equivalent to $\big(1 - o(1)\big)$ and that for large enough n, we have $\big(1 \pm o(1)\big) \leq 2$. Consequently, for sufficiently large n, we obtain $\mathbb{E}\big[|V(\mathcal{R})|\big] \leq 1/2 \cdot \tau \log^{(2)}(n)$. Since $k > \tau \log\log(n)$, it follows that $g(n) = 1/(1+\varepsilon) \cdot k$ is a valid upper bound on $\mathbb{E}\big[|V(\mathcal{R})|\big]$ for any $\varepsilon \in (0,1)$. Therefore, we can apply the Chernoff bound in Corollary 2.2 to conclude that

$$\begin{aligned}\Pr\Big[\big|V(\mathcal{R})\big| \geq k\Big] &\leq e^{-\varepsilon^2/3 \cdot g(n)} \\ &= e^{-\varepsilon^2/(3(1+\varepsilon)) \cdot k}.\end{aligned}$$

Choosing $\varepsilon = 1/2$ then yields the claim. □

We can now bound the expected number of vertices in all large narrow runs.

Lemma 7.12. *Let G be a hyperbolic random graph on n vertices. Then, the expected number of vertices in all large narrow runs is bounded by*

$$\mathbb{E}[N] = O\left(\frac{\tau \cdot n \cdot \log^{(2)}(n)}{\gamma(n,\tau)\log(n)^{\tau/18}}\right).$$

Proof. Let n'' denote the total number of narrow runs. We can compute the number of vertices in all large narrow runs, by summing over all narrow runs $\mathcal{R}_1, \dots, \mathcal{R}_{n''}$ and discarding the ones that are not large. That is,

$$N = \sum_{i=1}^{n''} \big|V(\mathcal{R}_i)\big| \cdot \mathbb{1}_{|V(\mathcal{R}_i)| > \tau \log^{(2)}(n)}.$$

Consequently, the expected value of N is given by

$$\begin{aligned}\mathbb{E}[N] &= \sum_{i=1}^{n''} \mathbb{E}\Big[\big|V(\mathcal{R}_i)\big| \cdot \mathbb{1}_{|V(\mathcal{R}_i)| > \tau \log^{(2)}(n)}\Big] \\ &= \sum_{i=1}^{n''} \sum_{k=\tau \log^{(2)}(n)+1}^{n} k \cdot \Pr\Big[\big|V(\mathcal{R}_i)\big| = k\Big].\end{aligned}$$

Using Lemma 7.11, we can now obtain a valid upper bound on $\Pr\big[|V(\mathcal{R}_i)| = k\big]$ for all $i \in [n'']$. Furthermore, the number of narrow runs n'' is bounded by the

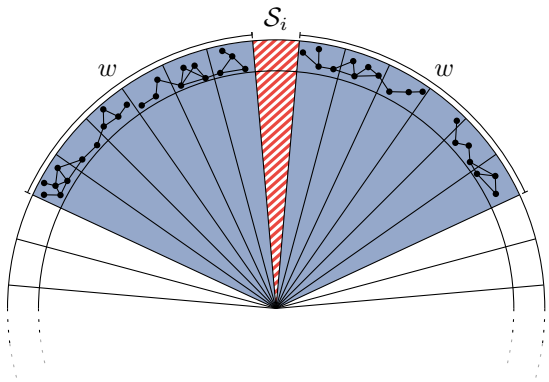


Figure 7.4: The random variable S_i indicates whether $\mathcal{S}_i$ contains any vertices. Changing S_i from 0 to 1 or vice versa merges two narrow runs or splits a wide run into two narrow ones, respectively. If all vertices were placed in the blue area, moving a single vertex in or out of $\mathcal{S}_i$ may change the number of vertices in large narrow runs by n.

number of sectors n'. Therefore, we obtain

$$\mathbb{E}[N] \leq n' \sum_{k=\tau \log^{(2)}(n)+1}^{n} k \cdot e^{-k/18}.$$

To get an upper bound, we replace the sum with an integral, which yields

$$\begin{aligned}\mathbb{E}[N] &\leq n' \int_{\tau \log^{(2)}(n)}^{n} k e^{-\frac{k}{18}} \, \mathrm{d}k \\ &\leq n' \left[18e^{-\tau/18 \log^{(2)}(n)}(\tau \log^{(2)}(n) + 18) - 18e^{-n/18}(n+18)\right] \\ &\leq 18n' \cdot \frac{\tau \log^{(2)}(n) + 18}{\log(n)^{\tau/18}},\end{aligned}$$

where the last inequality holds since $e^{-n/18}(n+18) \geq 0$. Finally, substituting $n' = 2n/\gamma(n,\tau)(1\pm o(1))$ (Lemma 7.3) and more simplification yield the claim. □

Concentration for the Number of Vertices in Large Narrow Runs. To show that the actual number of vertices in large narrow runs N is not much larger than the expected value, we consider N as a function of n independent random

variables $P_1, \dots, P_n$ representing the positions of the vertices in the hyperbolic disk. In order to show that N does not deviate much from its expected value with high probability, we would like to apply the method of bounded differences, which builds on the fact that N satisfies the bounded differences condition, i.e., that changing the position of a single vertex does not change N by much. Unfortunately, this change is not small in general.

In the worst case, there is a wide run $\mathcal{R}$ that contains all vertices and a sector $\mathcal{S}_i \subseteq \mathcal{R}$ contains only one of them. Moving this vertex out of $\mathcal{S}_i$ may split the run into two narrow runs (see Figure 7.4). These still contain n vertices, which corresponds to the change in N. However, this means that $\mathcal{R}$ consists of only few sectors (since it can be split into two narrow runs) and that all vertices lie within the corresponding (small) area of the disk. Since the vertices of the graph are distributed uniformly, this is very unlikely. To take advantage of this, we apply the method of *typical* bounded differences (see Section 2.2.3), which allows us to milden the effects of the change in the unlikely worst case and to focus on the typically smaller change of N instead. Formally, we represent the typical case using an event A denoting that each run of length at most $2w+1$ contains at most $\mathrm{O}\big(\log(n)\big)$ vertices. We now show that A occurs with probability $1-\mathrm{O}\big(n^{-c}\big)$ for any constant c, which proves that the atypical case is very unlikely.

Lemma 7.13. *Let G be a hyperbolic random graph. Then, each run of length at most $2w+1$ contains at most $\mathrm{O}\big(\log(n)\big)$ vertices with probability $1-\mathrm{O}\big(n^{-c}\big)$ for any constant c.*

Proof. We show that the probability for a single run $\mathcal{R}$ of at most $2w+1$ sectors to contain more then $\mathrm{O}\big(\log(n)\big)$ vertices is $\mathrm{O}\big(n^{-c_1}\big)$ for any constant c_1. Since there are at most $n' = \mathrm{O}(n)$ runs, applying the union bound and choosing $c_1 = c+1$ then yields the claim.

Recall that we divided the disk into n' sectors of equal width. Since the angular coordinates of the vertices are distributed uniformly at random, the probability for a given vertex to a lie in $\mathcal{R}$ is given by

$$\mu(\mathcal{R}) \leq \frac{2w+1}{n'} = \frac{2e^{\gamma(n,\tau)}\log^{(3)}(n)+1}{n'}.$$

By Lemma 7.3 the total number of sectors is given as $n' = 2n/\gamma(n,\tau)\cdot\big(1\pm\mathrm{o}(1)\big)$.

Consequently, we can compute the expected number of vertices in $\mathcal{R}$ as

$$\begin{aligned}\mathbb{E}\Big[\big|V(\mathcal{R})\big|\Big] &\leq n\mu(\mathcal{R}) \\ &= \Big(e^{\gamma(n,\tau)}\log^{(3)}(n) + 1/2\Big)\gamma(n,\tau)\big(1 \pm o(1)\big).\end{aligned}$$

Substituting $\gamma(n,\tau) = O\big(\log(\log^{(2)}(n)/\log^{(3)}(n)^2)\big)$, we can derive that

$$\begin{aligned}\mathbb{E}\Big[\big|V(\mathcal{R})\big|\Big] &\leq O\left(\frac{\log^{(2)}(n)}{\log^{(3)}(n)^2}\log^{(3)}(n)\cdot\log\left(\frac{\log^{(2)}(n)}{\log^{(3)}(n)^2}\right)\right) \\ &= O\Big(\log^{(2)}(n)\Big).\end{aligned}$$

Consequently, it holds that $g(n) = c_2\log(n)$ is a valid upper bound for any $c_2 > 0$ and large enough n. Therefore, we can apply the Chernoff bound in Corollary 2.2 to conclude that the probability for the number of vertices in $\mathcal{R}$ to exceed $g(n)$ is at most

$$\begin{aligned}\Pr\Big[\big|V(\mathcal{R})\big| \geq (1+\varepsilon)g(n)\Big] &\leq e^{-\varepsilon^2/3\cdot g(n)} \\ &= n^{-c_2\varepsilon^2/3}.\end{aligned}$$

As a consequence, c_2 can be chosen sufficiently large such that we obtain $\Pr\big[|V(\mathcal{R})| \geq (1+\varepsilon)g(n)\big] = O(n^{-c_1})$ for any constant c_1. □

The method of typical bounded differences now allows us to focus on this case and to milden the impact of the worst case changes as they occur with small probability. Consequently, we can show that the number of vertices in large narrow runs is sublinear with high probability.

Lemma 7.14. *Let G be a hyperbolic random graph on n vertices. Then, with high probability, the number of vertices in large narrow runs is bounded by*

$$N = O\left(\frac{\tau\cdot n\cdot\log^{(2)}(n)}{\gamma(n,\tau)\log(n)^{\tau/18}}\right).$$

Proof. Recall that the expected number of vertices in all large narrow runs is given by Lemma 7.12. Consequently, we can choose $c > 0$ large enough, such

that for sufficiently large n we obtain a valid upper bound on $\mathbb{E}[N]$ by choosing

$$g(n) = \frac{c \cdot \tau \cdot n \cdot \log^{(2)}(n)}{\gamma(n,\tau)\log(n)^{\tau/18}}.$$

In order to show that N does not exceed $g(n)$ by more than a constant factor with high probability, we apply the method of typical bounded differences (Corollary 2.8). To this end, we consider the typical event A, denoting that each run of at most $2w+1$ sectors contains at most $\mathrm{O}\big(\log(n)\big)$ vertices, and it remains to determine the parameters $\Delta_i^A \leq \Delta_i$ with which N satisfies the typical bounded differences condition with respect to A (see Equation (2.3)). Formally, we have to show that for all $i \in [n]$

$$\big|N(P_1,\ldots,P_i,\ldots,P_n) - N(P_1,\ldots,P_i',\ldots,P_n)\big| \leq \begin{cases} \Delta_i^A, \text{if } (P_1,\ldots,P_i,\ldots,P_n) \in A, \\ \Delta_i, \text{ otherwise.} \end{cases}$$

As argued before, changing the position P_i of vertex i to P_i' may result in a change of n in the worst case. Therefore, $\Delta_i = n$ is a valid bound for all $i \in [n]$. To bound the Δ_i^A, we have to consider the following situation. We start with a set of positions such that all runs of $2w+1$ sectors contain at most $\mathrm{O}\big(\log(n)\big)$ vertices and we want to bound the change in N when changing the position P_i of a single vertex i. In this case, splitting a wide run or merging two narrow runs can only change N by $\mathrm{O}\big(\log(n)\big)$. Consequently, we can choose $\Delta_i^A = \mathrm{O}\big(\log(n)\big)$ for all $i \in [n]$. By Corollary 2.8 we can now bound the probability that N exceeds $g(n)$ by more than a constant factor c_1 as

$$\Pr\big[N > c_1 g(n)\big] \leq e^{-((c_1-1)g(n))^2/(2\Delta)} + \Pr[\neg A] \cdot \sum_{i\in[n]} \frac{1}{\varepsilon_i},$$

for any $\varepsilon_1,\ldots,\varepsilon_n \in (0,1]$ and $\Delta = \sum_{i\in[n]}(\Delta_i^A + \varepsilon_i(\Delta_i - \Delta_i^A))^2$. By substituting the previously determined Δ_i^A and Δ_i, as well as, choosing $\varepsilon_i = 1/n$ for all $i \in [n]$, we obtain

$$\Delta = \mathrm{O}\Big(n \cdot \big(\log(n) + 1/n \cdot (n - \log(n))\big)^2\Big) = \mathrm{O}\Big(n \cdot \log(n)^2\Big).$$

Thus,

$$\begin{aligned}
&\Pr\big[N > c_1 g(n)\big] \\
&\quad \leq \exp\left(-\Theta\left(n^2 \cdot \left(\frac{\log^{(2)}(n)}{\gamma(n,\tau)\log(n)^{\tau/18}}\right)^2\right) \cdot \frac{1}{\mathrm{O}\big(n\log(n)^2\big)}\right) + \Pr[\neg A] \cdot \sum_{i\in[n]} \frac{1}{\varepsilon_i} \\
&\quad = \exp\left(-\Omega\left(n \cdot \left(\frac{\log^{(2)}(n)}{\gamma(n,\tau)\log(n)^{1+\tau/18}}\right)^2\right)\right) + \Pr[\neg A] \cdot \sum_{i\in[n]} \frac{1}{\varepsilon_i} \\
&\quad = \exp\left(-\Omega\left(n \cdot \left(\frac{\log^{(2)}(n)}{\log^{(3)}(n)\log(n)^{1+\tau/18}}\right)^2\right)\right) + \Pr[\neg A] \cdot \sum_{i\in[n]} \frac{1}{\varepsilon_i},
\end{aligned}$$

where the last equality holds, since $\gamma(n,\tau) = \mathrm{O}\big(\log^{(3)}(n)\big)$. By further simplifying the exponent, we can derive that the first part of the sum is $\exp\big(-\omega\big(\log(n)\big)\big)$. It follows that

$$\Pr\big[N > c_1 g(n)\big] \leq n^{-c_2} + \Pr[\neg A] \cdot \sum_{i\in[n]} \frac{1}{\varepsilon_i}$$

holds for any $c_2 > 0$ and sufficiently large n. It remains to bound the second part of the sum. Since $\varepsilon_i = 1/n$ for all $i \in [n]$, we have

$$\Pr[\neg A] \sum_{i\in[n]} \frac{1}{\varepsilon_i} = \Pr[\neg A] \cdot n^2.$$

By Lemma 7.13 it holds that $\Pr[\neg A] = \mathrm{O}\big(n^{-c_3}\big)$ for any c_3. Consequently, we can choose c_3 such that $\Pr[\neg A] \cdot n^2 = \mathrm{O}\big(n^{-(c_3-2)}\big)$ for any c_3, which concludes the proof. □

7.3.3 The Complete Disk

In the previous subsections we determined the number of vertices that are greedily added to the vertex cover in the inner disk and outer band, respectively. Before proving our main theorem, we are now ready to prove a slightly stronger

version that shows how the parameter τ can be used to obtain a trade-off between approximation performance and running time.

Theorem 7.15. *Let G be a hyperbolic random graph on n vertices with power-law exponent $\beta = 2\alpha + 1$ and let $\tau > 0$ be constant. Given the radii of the vertices, an approximate vertex cover of G can be computed in time $\mathrm{O}\big(n \log(n) + m \log(n)^{\tau}\big)$, such that the approximation factor is $\big(1 + \mathrm{O}\big(\gamma(n, \tau)^{-\alpha}\big)\big)$ asymptotically almost surely.*

Proof. Running Time. We start by sorting the vertices of the graph in order of increasing radius, which can be done in time $\mathrm{O}\big(n \log(n)\big)$. Afterwards, we iterate them and perform the following steps for each encountered vertex v. We add v to the cover, remove it from the graph, and identify connected components of size at most $\tau \log\log(n)$ that were separated by the removal. The first two steps can be performed in time $\mathrm{O}(1)$ and $\mathrm{O}\big(\deg(v)\big)$, respectively. Identifying and solving small components is more involved. Removing v can split the graph into at most $\deg(v)$ components, each containing a neighbor u of v. Such a component can be identified by performing a breadth-first search (BFS) starting at u. Each BFS can be stopped as soon as it encounters more than $\tau \log\log(n)$ vertices. The corresponding subgraph contains at most $(\tau \log\log(n))^2$ edges. Therefore, a single BFS takes time $\mathrm{O}\big(\log\log(n)^2\big)$. Whenever a component of size at most $n_c = \tau \log\log(n)$ is found, we compute a minimum vertex cover for it in time $1.1996^{n_c} \cdot n_c^{\mathrm{O}(1)}$ [XN17]. Since $n_c^{\mathrm{O}(1)} = \mathrm{O}\big((e/1.1996)^{n_c}\big)$, this running time is bounded by $\mathrm{O}\big(e^{n_c}\big) = \mathrm{O}\big(\log(n)^{\tau}\big)$. Consequently, the time required to process each neighbor of v is $\mathrm{O}\big(\log(n)^{\tau}\big)$. Since this is potentially performed for all neighbors of v, the running time of this third step can be bounded by introducing an additional factor of $\deg(v)$. We then obtain the total running time $T(n, m, \tau)$ of the algorithm by taking the time for the initial sorting and adding the sum of the running times of the above three steps over all vertices, which yields

$$\begin{aligned} T(n, m, \tau) &= \mathrm{O}\big(n \log(n)\big) + \sum_{v \in V} \mathrm{O}(1) + \mathrm{O}\big(\deg(v)\big) + \deg(v) \cdot \mathrm{O}\big(\log(n)^{\tau}\big) \\ &= \mathrm{O}\big(n \log(n)\big) + \mathrm{O}\left(\log(n)^{\tau} \cdot \sum_{v \in V} \deg(v)\right) \\ &= \mathrm{O}\big(n \log(n) + m \log(n)^{\tau}\big). \end{aligned}$$

Approximation Ratio. As argued before, we obtain a valid vertex cover for the whole graph, if we take all vertices in V_{Greedy} together with a vertex cover C_{Exact} of $G[V_{\text{Exact}}]$. The approximation ratio of the resulting cover is then given by the quotient

$$\xi = \frac{|V_{\text{Greedy}}| + |C_{\text{Exact}}|}{|C_{\text{OPT}}|},$$

where C_{OPT} denotes an optimal solution. Since all components in $G[V_{\text{Exact}}]$ are solved optimally and since any minimum vertex cover for the whole graph induces a vertex cover on $G[V']$ for any vertex subset $V' \subseteq V$, it holds that $|C_{\text{Exact}}| \leq |C_{\text{OPT}}|$. Therefore, the approximation ratio can be bounded by

$$\xi \leq 1 + \frac{|V_{\text{Greedy}}|}{|C_{\text{OPT}}|}.$$

To bound the number of vertices in V_{Greedy}, we add the number of vertices in the inner disk $\mathcal{D}_\rho$, as well as the numbers of vertices in the outer band that are contained in the area $\mathcal{W}$ that is covered by wide runs and the area $\mathcal{N}$ that is covered by large narrow runs. That is,

$$\xi \leq 1 + \frac{\left|V(\mathcal{D}_\rho)\right| + \left|V(\mathcal{W})\right| + \left|V(\mathcal{N})\right|}{|C_{OPT}|}.$$

Upper bounds on $|V(\mathcal{D}_\rho)|$, $|V(\mathcal{W})|$, and $|V(\mathcal{N})|$ that hold with high probability are given by Lemmas 7.2, 7.9 and 7.14, respectively. Furthermore, it was previously shown that the size of a minimum vertex cover on a hyperbolic random graph is $|C_{OPT}| = \Omega(n)$, asymptotically almost surely [CFR16, Theorems 4.10 and 5.8]. We obtain

$$\begin{aligned}\xi &= 1 + \mathrm{O}\left(\frac{1}{n}\left(\frac{n}{\gamma(n,\tau)^\alpha} + \frac{\tau^{3/4}\cdot n}{\log^{(2)}(n)^{1/4}\cdot\log^{(3)}(n)^{1/2}} + \frac{\tau\cdot n\cdot\log^{(2)}(n)}{\gamma(n,\tau)\log(n)^{\tau/18}}\right)\right)\\ &= 1 + \mathrm{O}\left(\frac{1}{\gamma(n,\tau)^\alpha} + \frac{\tau^{3/4}}{\log^{(2)}(n)^{1/4}\cdot\log^{(3)}(n)^{1/2}} + \frac{\tau\cdot\log^{(2)}(n)}{\gamma(n,\tau)\log(n)^{\tau/18}}\right).\end{aligned}$$

Since $\gamma(n,\tau) = \mathrm{O}\left(\log^{(3)}(n)\right)$, the first summand dominates asymptotically. □

We are now ready to prove our main theorem in this chapter, which we restate for the sake of readability.

Theorem 7.1. *Let G be the giant component of a hyperbolic random graph. Given the radii of the vertices, an approximate vertex cover of G can be computed in time $\mathrm{O}\big(m \log(n)\big)$, such that the approximation ratio is $\big(1 + \mathrm{o}(1)\big)$ asymptotically almost surely.*

Proof. By Theorem 7.15 we can compute an approximate vertex cover in time $\mathrm{O}\big(n \log(n) + m \log(n)^{\tau}\big)$, such that the approximation factor is $1 + \mathrm{O}\big(\gamma(n, \tau)^{-\alpha}\big)$, asymptotically almost surely. By choosing $\tau = 1$ we get $\gamma(n, 1) = \omega(1)$, which yields an approximation factor of $\big(1 + \mathrm{o}(1)\big)$, since $\alpha \in (1/2, 1)$. Additionally, the bound on the running time can be simplified to $\mathrm{O}\big(n \log(n) + m \log(n)\big)$. The claim then follows since we assume the graph to be connected, which implies that the number of edges is $m = \Omega(n)$. □

7.4 Empirical Evaluation

It remains to evaluate how well the predictions of our analysis on hyperbolic random graphs translate to real-world networks. According to the model, vertices near the center of the disk can likely be added to the vertex cover safely, while vertices near the boundary need to be treated more carefully (see Section 7.2). Moreover, it predicts that these boundary vertices can be found by identifying small components that are separated when removing vertices near the center. Due to the correlation between the radii of the vertices and their degrees, as explained in Section 3.3.3, this points to a natural extension of the standard greedy approach: While iteratively adding the vertex with the largest degree to the cover, small separated components are solved optimally. To evaluate how this approach compares to the standard greedy algorithm, we measured the approximation ratios on the largest connected component of a selection of 42 real-world networks from several network datasets [Kun13; RA15]. The results of our empirical analysis are summarized in Figure 7.5.

Our experiments confirm that the standard greedy approach already yields close to optimal approximation ratios on all networks, as it has been observed previously [SGS13]. In fact, the "worst" approximation ratio is only 1.049 for the network `dblp-cite`. The average lies at just 1.009.

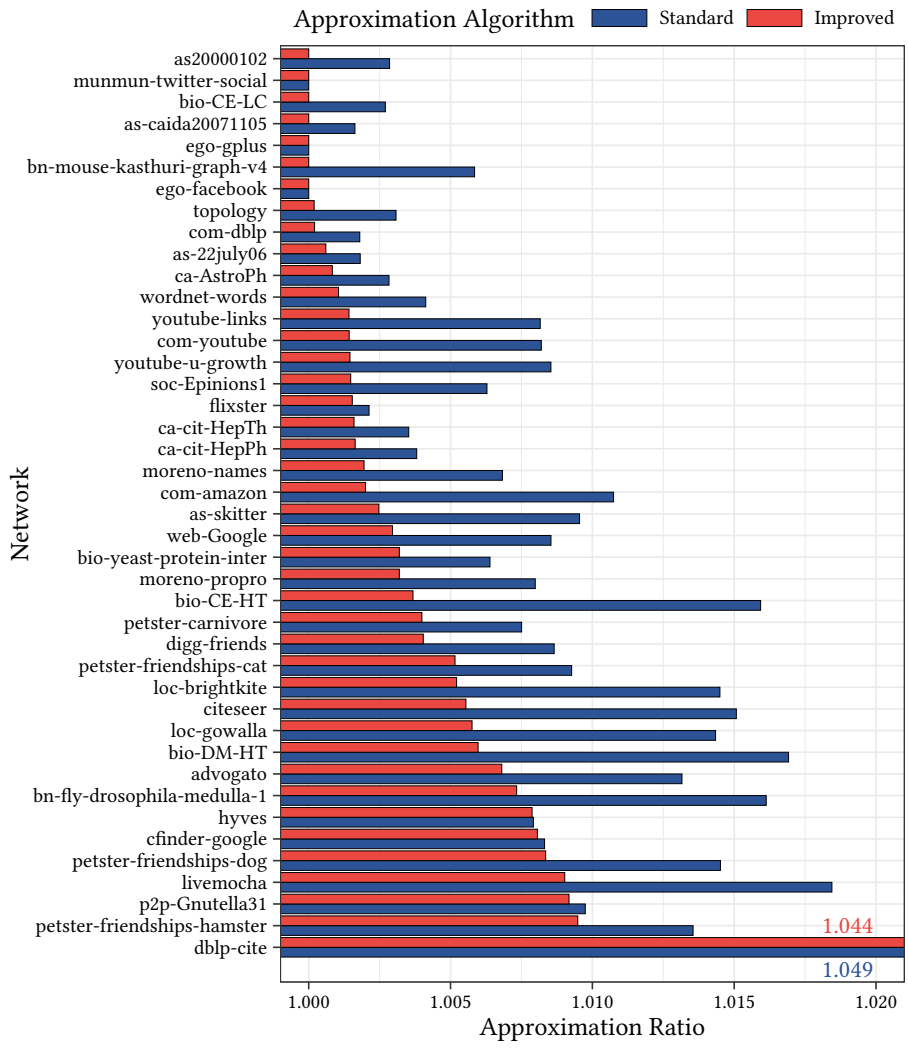


Figure 7.5: Approximation ratios obtained using the standard greedy approach (blue) and our improved version (red) on a selection of real-world networks. The parameter adjusting the component size threshold was chosen as $\tau = 10$. For the sake of readability the bars denoting the ratios for the `dblp-cite` network were cropped and the actual values written next to them.

Clearly, our adapted greedy approach performs at least as well as the standard greedy algorithm. In fact, for $\tau = 1$ the sizes of the components that are solved optimally on the considered networks are at most 3. For components of this size the standard greedy approach performs optimally. Therefore, the approximation performances of the standard and the adapted greedy match in this case. However, the adapted greedy algorithm allows for improving the approximation ratio by increasing the size of the components that are solved optimally. In our experiments, we chose $10\lceil \log \log(n) \rceil$ as the component size threshold, which corresponds to setting $\tau = 10$. The resulting impact can be seen in Figure 7.6, which shows the error of the adapted greedy compared to the one of the standard greedy algorithm. This relative error is measured as the fraction of the number of vertices by which the adapted greedy and the standard approach exceed an optimum solution. That is, a relative error of 0.5 indicates that the adapted greedy halved the number of vertices by which the solution of the standard greedy exceeded an optimum. Moreover, a relative error of 0 indicates that the adapted greedy found an optimum when the standard greedy did not. The relative error is omitted if the standard greedy already found an optimum, i.e., there was no error to improve on. For more than 69% of the considered networks (29 out of 42) the relative error is at most 0.5 and the average relative error is 0.39. Since the behavior of the two algorithms only differs when it comes to small separated components, this indicates that the predictions of the model that led to the improvement of the standard greedy approach do translate to real-world networks. In fact, the average approximation ratio obtained using the standard greedy algorithm is reduced from 1.009 to 1.004 when using the adapted greedy approach.

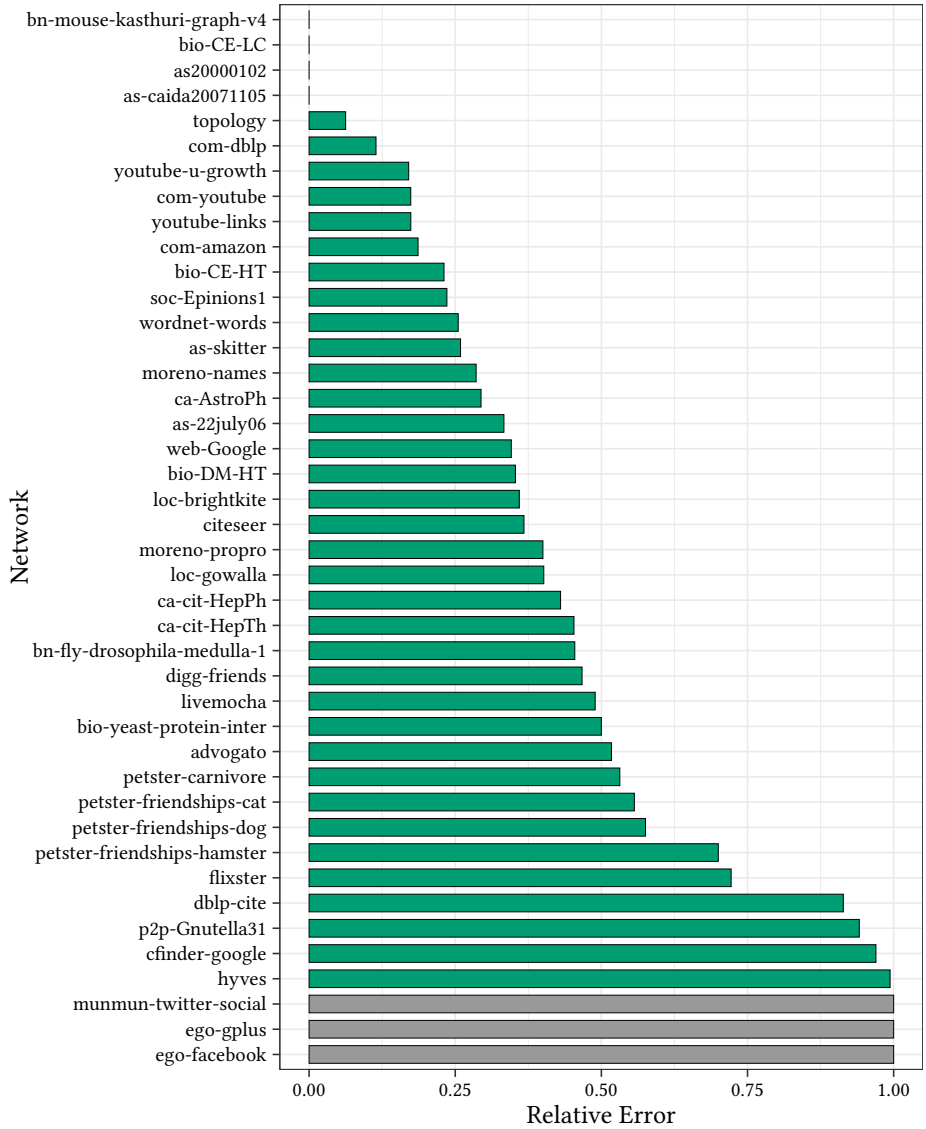


Figure 7.6: Relative error of the improved greedy compared to the standard approach. The parameter adjusting the component size threshold was chosen as $\tau = 10$. Gray bars indicate that no error could be determined since the standard approach found an optimum already.

8 Conclusions and Outlook

Throughout the thesis, we have examined algorithms on networks that feature an underlying hyperbolic geometry. To this end, we first introduced strongly hyperbolic unit disk graphs as a subclass of hyperbolic unit disk graphs and identified the commonly used version of hyperbolic random graphs as a special case thereof. We then analyzed algorithmic solutions to several fundamental problems in graph theory, including routing, path finding, and the vertex cover problem. For each of them, empirical observations have shown that solutions can often be obtained much more efficiently than one would expect in the worst case, which indicates that real networks are inherently different from worst-case instances with respect to certain properties. By analyzing networks through the lens of hyperbolic geometry, we captured these properties and were able to reduce the gap between observed performance and worst-case bounds for all considered problems.

Naturally, all our analyses exploited the underlying geometry. To this end, we discretized the considered hyperbolic disk using various geometric shapes such as sectors and annuli (i.e., bands). Most commonly, distinguishing between an inner region of the disk and an outer one, in order to consider high-degree vertices and ones with smaller degree separately proved to be of essence. In combination with stochastic tools, in particular certain concentration bounds like the method of typically bounded differences, and the use of a poissonized version of hyperbolic random graphs, this allowed us to uncover interesting properties of the model, which can be exploited algorithmically. These include a bound on the size of the 2-neighborhood of a vertex, i.e., the number of vertices at distance two from it, which we then utilized to bound the number of edges explored in a breadth-first search (see Chapter 5). Moreover, we showed that high-degree vertices are likely to dominate other vertices, i.e., they are neighbors to them and all their neighbors, and proved that removing high-degree vertices from a hyperbolic random graph yields a remainder of small pathwidth. Combined, both properties build the foundation of the proof that optimal vertex covers can be obtained in polynomial time on such graphs (see Chapter 6). Additionally,

when removing more vertices, hyperbolic random graphs are decomposed into small components, which we utilized in the proof that a good vertex cover approximation can be computed efficiently on such graphs (see Chapter 7). Our empirical evaluations on real-world networks then gave evidence that these properties can also be observed in practice.

In conclusion, by utilizing networks with an underlying hyperbolic geometry, we were able

- to understand and explain prior empirical observations from a theoretical point of view,
- to derive network properties, which we then also observed in real-world networks, and
- to utilize these insights in order to improve an existing algorithm, which lead to a performance gain in practice.

This shows that exploiting the connection between complex real-world networks and hyperbolic geometry yields a powerful tool in the context of average-case analysis.

To further utilize this connection, our research could be extended in several ways that basically involve three general approaches: further algorithmic analyses, improving the framework, and utilizing hyperbolic geometry in other contexts.

Further Algorithmic Analyses. While we have demonstrated how to make use of the underlying hyperbolic geometry in several algorithmic analyses in the context of routing, path finding, and the famous vertex cover problem, an obvious question is whether the approach can be applied to other problems as well. An example would be the *graph reconstruction problem*. There, we are given the vertex set of a graph, but are missing the information about the edges. Instead, there is some kind of oracle that can be queried to obtain information about the topology of the network. The goal then is to uncover the edge set of the graph by using the oracle efficiently. An exemplary real-world application is the internet, which is a decentralized network, meaning there is no single instance that knows the whole graph at a given point in time. Therefore, graph reconstruction approaches are applied to get an understanding of its structure. In recent work, a simple algorithm was developed that solves the problem on

random d-regular graphs (every vertex has the same degree d), requiring only few queries to the oracle [MZ21]. Since the algorithm exploits the *locally tree-like* property of such networks, the authors conjecture that the method is also applicable on scale-free graphs. In particular, investigating the performance of their approach (and the graph reconstruction problem in general) on hyperbolic random graphs may yield insights that translate to the key application of the problem on the internet, as prior work has shown that the internet fits well into hyperbolic space [BPK10].

Additionally, it would be interesting to study algorithms for computing the diameter of a network, which are often based on breadth-first searches (as studied in Chapter 5) and observed to work well in practice [Cre+10]. Moreover, recent results in the context of finding the *connected components* in a graph [Bro+21], as well as an empirical study of heuristics for the *vertex connectivity problem* [FY21] further motivate the analysis of algorithms on hyperbolic random graphs to theoretically explain good performance on real-world networks.

Improving the Framework. As we have seen, network models with underlying hyperbolic geometry yield good representations of real-world networks. However, the real world is typically rather noisy and does not perfectly match the rules that are used to generate hyperbolic random graphs. To accommodate for this, hyperbolic random graphs have also been studied in a noisy setting where, with some small probability, distant vertices are connected and close vertices are not connected [Kri+10]. Similarly, the hyperbolic unit disk graph variant of Kisfaludi-Bak [Kis20] is also studied in a noisy version. Consequently, it would be interesting to also study (strongly) hyperbolic unit disk graphs in a noisy variant. To the best of our knowledge, algorithmic results in noisy versions of these models have not been obtained so far. However, a promising approach was proposed in the context of Boolean satisfiability [Blä+21b]. There, the authors propose a model for generating formulas with an underlying geometry, which they also study in a noisy setting. More precisely, the building blocks of a formula, *variables* and *clauses*, are mapped into a geometric space and then variables are assigned to clauses if their distance is sufficiently small. The introduced noise then allows for assignments where the distances are large, or inhibits assignment, even if the distance is small. Their analysis of the model builds on the concept of *nice clauses*, i.e., clauses where the assignment turned out just as it would have in a non-noisy setting. In a similar fashion, one could consider

nice vertices in noisy variants of network models with underlying hyperbolic geometry. These are vertices whose neighborhood is the same as it would have been in the non-noisy setting. If the noise is not too large, i.e., if there are sufficiently many nice vertices, the properties we utilized in the analyses throughout the thesis may then still hold. For example, in the vertex cover problem, a vertex needs to dominate only one other vertex in order to be eligible for removal by the reduction rule. Thus, even if the majority of the vertices are not nice, it suffices if high-degree vertices are and have at least one nice neighbor in their dominance area.

In addition to the noisy variant of the model, other extensions may be of interest. Recent work has started looking into a weighted version of hyperbolic random graphs [KL20], and while the current model uses the two-dimensional hyperbolic plane as the underlying space, there are adaptations that extend to higher dimensions [YR20]. In particular, the *geometric inhomogeneous random graph (GIRG)* model seems to be a very approachable generalization of hyperbolic random graphs that easily generalizes to higher dimensionality [BKL19].

Utilizing Hyperbolic Geometry in Other Contexts. While defining the model using two-dimensional hyperbolic space yields several advantages, such as the mathematical accessibility and the ability to visualize the graphs, it is questionable whether the intrinsic dimensionality of complex real-world networks is always this low. In fact, when researching the connection between networks and an underlying hyperbolic geometry in the context of *deep learning*, recent work suggests that data often fits better in higher dimensions (in the order of 10 to more than 100) [Pen+21]. This leaves a huge gap in our understanding. Investigating this discrepancy may help in improving our tools in order to obtain even better representations of real-world networks. The idea of researching the dimensionality of a graph is not new (see, e.g., [EHT65]). However, prior work has mainly focused on the network dimension in Euclidean geometry [KL07; RRŠ89]. There, one way to empirically measure the dimensionality of a graph would be to perform several breadth-first searches and to observe how the size of the search space grows with each step. In a Euclidean random graph, where vertices are distributed uniformly in the hypercube $[0, 1]^d$, one would expect that this growth roughly matches how the volume of a d-ball grows with increasing radius. In hyperbolic geometry, a similar approach may face difficulties due to the heterogeneity of the networks and the exponential expansion of hyperbolic

space. Performing a breadth-first search from a high-degree vertex behaves rather differently compared to starting from a vertex with small degree. Additionally, since the diameter of such networks is typically small (e.g., logarithmic), the whole graph is explored in only very few steps, potentially making it difficult to distinguish between different exponentially growing functions. However, on the examined networks it may suffice to only consider lower-degree vertices. As we have seen in Chapter 6, removing high-degree vertices leaves a remainder with a path-like structure. We conjecture that this extends to grid-like structures of higher dimensionality, when considering hyperbolic random graphs in higher dimensions. In that case we may still be able to measure the network dimension by constraining the breadth-first searches to the low-degree portions of the graph.

Another approach to measuring the dimensionality of a network are *embeddings*. There, the idea is to map the vertices of a graph into some metric space, such that the distances in the graph are reflected by the distances in the metric. A prominent example is the embedding of an internet topology into the hyperbolic plane [BPK10]. Moreover, being of fundamental interest in the field of knowledge representation, the problem has gained a lot of attention in the deep learning community [Pen+21]. For hyperbolic random graphs, several embedding algorithms have been proposed [AMA16; Blä+16; Mus+17; PAK15; PPK15] and we have previously shown that the well-known and versatile *force-directed embedding* approach can be adapted to work in hyperbolic space [BFK21b]. While this technique can be easily extended to work in higher dimensions, the standard approach is inherently slow, requiring further extensions to improve its running time. On the other hand, a maximum likelihood embedder has proven to be very efficient in computing good embeddings of large networks in the two-dimensional hyperbolic plane [Blä+16]. Extending such a performant embedder to work in higher dimensions may allow for embedding a given network into spaces of increasing dimensionality, until further increases do not improve the quality of the embedding significantly, which indicates that this is the smallest dimension that the network fits in well.

Finally, we can extend the hyperbolic random graph model to produce other kinds of networks, like trees. Insights in the context of epidemiology suggest that infections in a network spread heterogeneously, in the sense that there are a few *super spreaders* (i.e., high-degree vertices) that are responsible for the vast majority of all infections [GM05]. Consequently, having a model that

produces heterogeneous trees may provide useful to represent and analyze epidemiologic transmission trees [HWR16]. Following the success of utilizing hyperbolic geometry to represent real-world networks, in such a model we may start with a hyperbolic random graph, weight the edges using the hyperbolic distances between the vertices, and obtain a tree by removing edges such that the sum of the edge weights is minimized. Alternatively, instead of starting with containing all edges whose length is below a certain threshold, one could consider the complete graph on the sampled point set and reduce the edge set in the same way. This would yield a random version of the hyperbolic equivalent to *Euclidean minimum spanning trees* (see, e.g., [SH75]). An efficient algorithm for the generation of such trees consists of computing the *Delaunay triangulation* of the points, which reduces the size of the relevant edge set from quadratic to linear, and subsequently applying a minimum-spanning-tree algorithm to obtain the desired graph. Unfortunately, the polar-coordinate model of the hyperbolic plane is not as well suited for geometric applications as other models and conversion between different representations of hyperbolic space can lead to numerical issues. However, our preliminary work shows that the *Voronoi diagram*, i.e., the dual graph of the Delaunay triangulation, can be computed natively in the polar-coordinate model [FKS21]. Furthermore, preliminary experiments indicate that the trees computed this way do feature a power-law degree distribution and we believe that further theoretical and empirical analyses may prove worthwhile.

Bibliography

[AB02] Réka Albert and Albert-László Barabási. **Statistical Mechanics of Complex Networks**. *Reviews of Modern Physics* 74 (2002), 47–97. DOI: 10.1103/RevModPhys.74.47 (see page 8).

[Abr+08] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, Noam Nisan, and Mikkel Thorup. **Compact Name-Independent Routing with Minimum Stretch**. *ACM Transactions on Algorithms* 4:3 (2008). DOI: 10.1145/1367064.1367077 (see page 50).

[Abu+04] Faisal N. Abu-Khzam, Rebecca L. Collins, Michael R. Fellows, Michael A. Langston, W. Henry Suters, and Christopher T. Symons. **Kernelization Algorithms for the Vertex Cover Problem: Theory and Experiments**. In: *Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics*. 2004, 62–69 (see page 109).

[ACL01] William Aiello, Fan Chung, and Linyuan Lu. **A Random Graph Model for Power Law Graphs**. *Experimental Mathematics* 10:1 (2001), 53–66. DOI: 10.1080/10586458.2001.10504428 (see pages 4, 52).

[ACL12] Eric Angel, Romain Campigotto, and Christian Laforest. **Implementation and Comparison of Heuristics for the Vertex Cover Problem on Huge Graphs**. In: *International Symposium on Experimental Algorithms*. 2012, 39–50. DOI: 10.1007/978-3-642-30850-5_5 (see page 133).

[ADF12] Patrizio Angelini, Giuseppe Di Battista, and Fabrizio Frati. **Succinct Greedy Drawings Do Not Always Exist**. *Networks* 59:3 (2012), 267–274. DOI: 10.1002/net.21449 (see page 51).

[AG13] Rachit Agarwal and Philip Brighten Godfrey. **Brief Announcement: A Simple Stretch 2 Distance Oracle**. In: *ACM Symposium on Principles of Distributed Computing (PODC)*. Ed. by Panagiota Fatourou and Gadi Taubenfeld. 2013, 110–112. DOI: 10.1145/2484239.2484277 (see page 50).

[AGH11] Rachit Agarwal, Philip Brighten Godfrey, and Sariel Har-Peled. **Approximate Distance Queries and Compact Routing In Sparse Graphs**. In: *2011 Proceedings IEEE INFOCOM*. 2011, 1754–1762. DOI: 10.1109/INFCOM.2011.5934973 (see pages 50, 52).

[AGM04] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. **Routing with Improved Communication-Space Trade-Off**. In: *18th International Symposium on Distributed Computing (DISC)*. Vol. 3274. 2004, 305–319. DOI: 10.1007/978-3-540-30186-8_22 (see page 50).

[AGM06] Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. **On Space-Stretch Trade-Offs: Upper Bounds**. In: *Annual ACM Symposium on Parallelism in Algorithms and Architectures*. Vol. 2006. 2006, 217–224. DOI: 10.1145/1148109.1148144 (see page 50).

[AI16] Takuya Akiba and Yoichi Iwata. **Branch-and-Reduce Exponential/FPT Algorithms in Practice: A Case Study of Vertex Cover**. *Theoretical Computer Science* 609 (2016), 211–225. DOI: 10.1016/j.tcs.2015.09.023 (see pages 2, 10, 110, 111, 133).

[AKP94] Baruch Awerbuch, Shay Kutten, and David Peleg. **On Buffer-economical Store-and-forward Deadlock Prevention**. *IEEE Transactions on Communications* 42:11 (1994), 2934–2937. DOI: 10.1109/26.328973 (see page 57).

[Alb05] Réka Albert. **Scale-Free Networks in Cell Biology**. *Journal of Cell Science* 118:21 (2005), 4947–4957. DOI: 10.1242/jcs.02714 (see page 8).

[AMA16] Gregorio Alanis-Lobato, Pablo Mier, and Miguel A. Andrade-Navarro. **Manifold Learning and Maximum Likelihood Estimation For Hyperbolic Network Embedding**. *Applied Network Science* 1:10 (2016), 1–14. DOI: 10.1007/s41109-016-0013-0 (see page 169).

[AP92] Baruch Awerbuch and David Peleg. **Routing with Polynomial Communication-Space Trade-Off**. *SIAM Journal on Discrete Mathematics* 5 (1992), 151–162. DOI: 10.1137/0405013 (see pages 50, 57).

[Are+] Alexandre Arenas, Albert-László Barabási, Vladimir Batagelj, Andrej Mrvar, Mark Newman, and Tore Opsahl. *Gephi Datasets*. https://github.com/gephi/gephi/wiki/Datasets (see page 129).

[Art+20] Igor Artico, Igor E. Smolyarenko, Veronica Vinciotti, and Ernst C. Wit. **How Rare Are Power-Law Networks Really?** *Proceedings of the Royal Society A* 476:2241 (2020), 20190742. DOI: 10.1098/rspa.2019.0742 (see page 4).

[ASK12] Takuya Akiba, Christian Sommer, and Ken-ichi Kawarabayashi. **Shortest-Path Queries for Complex Networks: Exploiting Low Tree-Width Outside the Core**. In: *Proceedings of the 15th International Conference on Extending Database Technology (EDBT)*. 2012, 144–155. DOI: 10.1145/2247596.2247614 (see page 71).

[Awe+90] Baruch Awerbuch, Amotz Bar-Noy, Nathan Linial, and David Peleg. **Improved Routing Strategies with Succinct Tables**. *Journal of Algorithms* 11:3 (1990), 307–341. DOI: 10.1016/0196-6774(90)90017-9 (see page 50).

[BA99] Albert-László Barabási and Réka Albert. **Emergence of Scaling in Random Networks**. *Science* 286:5439 (1999), 509–512. DOI: 10.1126/science.286.5439.509 (see page 4).

[Bac+12] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. **Four Degrees of Separation**. In: *4th Annual ACM Web Science Conference*. 2012, 33–42. DOI: 10.1145/2380718.2380723 (see page 6).

[BC06] Arthur Brady and Lenore Cowen, 119–128. In: *Workshop on Algorithm Engineering and Experiments (ALENEX)*. 2006. DOI: 10.1137/1.9781611972863.12 (see page 51).

[BFK16] Thomas Bläsius, Tobias Friedrich, and Anton Krohmer. **Hyperbolic Random Graphs: Separators and Treewidth**. In: *24th Annual European Symposium on Algorithms (ESA)*. 2016, 15:1–15:16. DOI: 10.4230/LIPIcs.ESA.2016.15 (see pages 41, 110).

[BFK18] Thomas Bläsius, Tobias Friedrich, and Anton Krohmer. **Cliques in Hyperbolic Random Graphs**. *Algorithmica* 80 (2018), 2324–2344. DOI: 10.1007/s00453-017-0323-3 (see page 41).

[BFK21a] Thomas Bläsius, Tobias Friedrich, and Maximilian Katzmann. **Efficiently Approximating Vertex Cover on Scale-Free Networks with Underlying Hyperbolic Geometry**. In: *29th Annual European Symposium on Algorithms (ESA)*. 2021, 20:1–20:15. DOI: 10.4230/LIPIcs.ESA.2021.20 (see page 133).

[BFK21b] Thomas Bläsius, Tobias Friedrich, and Maximilian Katzmann. **Force-Directed Embedding of Scale-Free Networks in the Hyperbolic Plane**. In: *19th International Symposium on Experimental Algorithms (SEA)*. 2021, 22:1–22:18. DOI: 10.4230/LIPIcs.SEA.2021.22 (see page 169).

[BFM13] Michel Bode, Nikolaos Fountoulakis, and Tobias Müller. **On the Giant Component of Random Hyperbolic Graphs**. In: *The Seventh European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB)*. 2013, 425–429. DOI: 10.1007/978-88-7642-475-5_68 (see page 40).

[BFM15] Michel Bode, N. Fountoulakis, and Tobias Müller. **On the Largest Component of a Hyperbolic Model of Complex Networks**. *Electronic Journal of Combinatorics* 22 (2015), 1–46. DOI: 10.37236/4958 (see pages 6, 40, 42).

[BFM16] Michel Bode, Nikolaos Fountoulakis, and Tobias Müller. **The Probability of Connectivity in a Hyperbolic Model of Complex Networks**. *Random Structures & Algorithms* 49:1 (2016), 65–94. DOI: 10.1002/rsa.20626 (see page 40).

[BKL19] Karl Bringmann, Ralph Keusch, and Johannes Lengler. **Geometric Inhomogeneous Random Graphs**. *Theoretical Computer Science* 760 (2019), 35–54. DOI: 10.1016/j.tcs.2018.08.014 (see pages 40, 41, 168).

[Blä+16] Thomas Bläsius, Tobias Friedrich, Anton Krohmer, and Sören Laue. **Efficient Embedding of Scale-Free Graphs in the Hyperbolic Plane**. In: *24th Annual European Symposium on Algorithms (ESA)*. 2016, 16:1–16:18. DOI: 10.4230/LIPIcs.ESA.2016.16 (see page 169).

[Blä+18] Thomas Bläsius, Cedric Freiberger, Tobias Friedrich, Maximilian Katzmann, Felix Montenegro-Retana, and Marianne Thieffry. **Efficient Shortest Paths in Scale-Free Networks with Underlying Hyperbolic Geometry**. In: *45th International Colloquium on Automata, Languages, and Programming (ICALP)*. 2018, 20:1–20:14. DOI: 10.4230/LIPIcs.ICALP.2018.20 (see page 71).

[Blä+19] Thomas Bläsius, Tobias Friedrich, Maximilian Katzmann, Ulrich Meyer, Manuel Penschuck, and Christopher Weyand. **Efficiently Generating Geometric Inhomogeneous and Hyperbolic Random Graphs**. In: *27th Annual European Symposium on Algorithms (ESA 2019)*. Vol. 144. 2019, 21:1–21:14. DOI: 10.4230/LIPIcs.ESA.2019.21 (see page 41).

[Blä+20] Thomas Bläsius, Tobias Friedrich, Maximilian Katzmann, and Anton Krohmer. **Hyperbolic Embeddings for Near-Optimal Greedy Routing**. *ACM Journal of Experimental Algorithmics* 25 (2020). DOI: 10.1145/3381751 (see page 51).

[Blä+21a] Thomas Bläsius, Philipp Fischbeck, Tobias Friedrich, and Maximilian Katzmann. **Solving Vertex Cover in Polynomial Time on Hyperbolic Random Graphs**. *Theory of Computing Systems* (2021). DOI: 10.1007/s00224-021-10062-9 (see page 109).

[Blä+21b] Thomas Bläsius, Tobias Friedrich, Andreas Göbel, Jordi Levy, and Ralf Rothenberger. **The Impact of Heterogeneity and Geometry on the Proof Complexity of Random Satisfiability**. In: *Symposium on Discrete Algorithms (SODA)*. 2021, 42–53 (see page 167).

[Blä+22] Thomas Bläsius, Cedric Freiberger, Tobias Friedrich, Maximilian Katzmann, Felix Montenegro-Retana, and Marianne Thieffry. **Efficient Shortest Paths in Scale-Free Networks with Underlying Hyperbolic Geometry**. *ACM Trans. Algorithms* 18 (2022), 19:1–19:32. DOI: 10.1145/3516483 (see page 71).

[Blä+23] Thomas Bläsius, Tobias Friedrich, Maximilian Katzmann, and Daniel Stephan. **Strongly Hyperbolic Unit Disk Graphs**. In: *40th International Symposium on Theoretical Aspects of Computer Science (STACS)*. Accepted for publication. 2023 (see pages 28, 47).

[BM06] Vladimir Batagelj and Andrej Mrvar. *Pajek datasets*. http://vlado.fmf.uni-lj.si/pub/networks/data/. 2006 (see page 129).

[BN16] Michele Borassi and Emanuele Natale. **KADABRA is an ADaptive Algorithm for Betweenness via Random Approximation**. In: *24th Annual European Symposium on Algorithms (ESA)*. 2016, 20:1–20:18. DOI: 10.4230/LIPIcs.ESA.2016.20 (see pages 9, 71–73, 108).

[Bol01] Béla Bollobás. **Random Graphs**. 2nd ed. Cambridge University Press, 2001. DOI: 10.1017/CBO9780511814068 (see page 3).

[BPK10] Marián Boguñá, Fragkiskos Papadopoulos, and Dmitri Krioukov. **Sustaining the Internet with Hyperbolic Mapping**. *Nature Communications* 1 (2010), 1–8. DOI: 10.1038/ncomms1063 (see pages 6, 48, 52, 167, 169).

[Bri+17] Karl Bringmann, Ralph Keusch, Johannes Lengler, Yannic Maus, and Anisur Rahaman Molla. **Greedy Routing and the Algorithmic Small-World Phenomenon**. In: *ACM Symposium on Principles of Distributed Computing*. 2017, 371–380. DOI: 10.1145/3087801.3087829 (see pages 41, 52).

[Bro+21] Gerth Stølting Brodal, Rolf Fagerberg, David Hammer, Ulrich Meyer, Manuel Penschuck, and Hung Tran. **An Experimental Study of External Memory Algorithms for Connected Components**. In: *19th International Symposium on Experimental Algorithms (SEA)*. 2021, 23:1–23:23. DOI: 10.4230/LIPIcs.SEA.2021.23 (see page 167).

[Cai94] Leizhen Cai. **NP-completeness of Minimum Spanner Problems**. *Discrete Applied Mathematics* 48:2 (1994), 187–194. DOI: 10.1016/0166-218X(94)90073-6 (see page 54).

[Cam+14] Miguel Camelo, Dimitri Papadimitriou, Lluís Fàbrega, and Pere Vilà. **Geometric Routing With Word-Metric Spaces**. *IEEE Communications Letters* 18:12 (2014), 2125–2128. DOI: 10.1109/LCOMM.2014.2364213 (see page 51).

[CF16a] Elisabetta Candellero and Nikolaos Fountoulakis. **Bootstrap Percolation and the Geometry of Complex Networks**. *Stochastic Processes and their Applications* 126:1 (2016), 234–264. DOI: 10.1016/j.spa.2015.08.005 (see page 41).

[CF16b] Elisabetta Candellero and Nikolaos Fountoulakis. **Clustering and the Hyperbolic Geometry of Complex Networks**. *Internet Mathematics* 12:1-2 (2016), 2–53. DOI: 10.1080/15427951.2015.1067848 (see page 41).

[CFR16] Ankit Chauhan, Tobias Friedrich, and Ralf Rothenberger. **Greed is Good for Deterministic Scale-Free Networks**. In: *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*. 2016, 33:1–33:15. DOI: 10.4230/LIPIcs.FSTTCS.2016.33 (see pages 134, 138, 160).

[Che+12] Wei Chen, Christian Sommer, Shang-Hua Teng, and Yajun Wang. **A Compact Routing Scheme and Approximate Distance Oracle for Power-Law Graphs**. *ACM Transactions on Algorithms* 9:1 (2012). DOI: 10.1145/2390176.2390180 (see pages 50, 52).

[CJ03] Liming Cai and David Juedes. **On the Existence of Subexponential Parameterized Algorithms**. *Journal of Computer and System Sciences* 67 (2003), 789–807. DOI: 10.1016/S0022-0000(03)00074-6 (see page 109).

[CKX10] Jianer Chen, Iyad A. Kanj, and Ge Xia. **Improved Upper Bounds for Vertex Cover**. *Theoretical Computer Science* 411:40 (2010), 3736–3756. DOI: 10.1016/j.tcs.2010.06.026 (see page 109).

[CL02a] Fan Chung and Linyuan Lu. **Connected Components in Random Graphs with Given Expected Degree Sequences**. *Annals of Combinatorics* 6:25 (2002), 125–145. DOI: 10.1007/PL00012580 (see page 4).

[CL02b] Fan Chung and Linyuan Lu. **The Average Distances in Random Graphs with Given Expected Degrees**. *Proceedings of the National Academy of Sciences* 99:25 (2002), 15879–15882. DOI: 10.1073/pnas.252631999 (see page 4).

[Coh98] Edith Cohen. **Fast Algorithms for Constructing t-Spanners and Paths with Stretch t**. *SIAM Journal on Computing* 28:1 (1998), 210–236. DOI: 10.1137/S0097539794261295 (see pages 48, 58).

[Cow01] Lenore J. Cowen. **Compact Routing with Minimum Stretch**. *Journal of Algorithms* 38:1 (2001), 170–183. DOI: 10.1006/jagm.2000.1134 (see page 50).

[Cre+10] Pierluigi Crescenzi, Roberto Grossi, Claudio Imbrenda, Leonardo Lanzi, and Andrea Marino. **Finding the Diameter in Real-World Graphs**. In: *18th Annual European Symposium on Algorithms (ESA)*. 2010, 302–313. DOI: 10.1007/978-3-642-15775-2_26 (see page 167).

[Cyg+15] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. **Parameterized Algorithms**. Springer, 2015. ISBN: 978-3-319-21274-6. DOI: 10.1007/978-3-319-21275-3 (see pages 2, 120, 126).

[DC02] Jesper Dall and Michael Christensen. **Random Geometric Graphs**. *Physical Review E* 66 (2002), 016121. DOI: 10.1103/PhysRevE.66.016121 (see page 6).

[Del+18] Holger Dell, Christian Komusiewicz, Nimrod Talmon, and Mathias Weller. **The PACE 2017 Parameterized Algorithms and Computational Experiments Challenge: The Second Iteration**. In: *12th International Symposium on Parameterized and Exact Computation (IPEC)*. 2018, 30:1–30:12. DOI: 10.4230/LIPIcs.IPEC.2017.30 (see page 2).

[DFH19] M. Ayaz Dzulfikar, Johannes K. Fichte, and Markus Hecher. **The PACE 2019 Parameterized Algorithms and Computational Experiments Challenge: The Fourth Iteration (Invited Paper)**. In: *14th International Symposium on Parameterized and Exact Computation (IPEC)*. 2019, 25:1–25:23. DOI: 10.4230/LIPIcs.IPEC.2019.25 (see page 2).

[DG02] Yon Dourisboure and Cyril Gavoille. **Improved Compact Routing Scheme for Chordal Graphs**. In: *International Symposium on Distributed Computing (DISC)*. 2002, 252–264. DOI: 10.1007/3-540-36108-1_17 (see page 50).

[DHZ96] Dorit Dor, Shay Halperin, and Uri Zwick. **All Pairs Almost Shortest Paths**. In: *37th Conference on Foundations of Computer Science*. 1996, 452–461. DOI: 10.1109/SFCS.1996.548504 (see page 50).

[Die17] Reinhard Diestel. **Graph Theory**. Springer Berlin, 2017. ISBN: 978-3-662-53621-6 (see page 11).

[Dor10] Sergey Dorogovtsev. **Lectures on Complex Networks**. Oxford University Press, Inc., 2010. ISBN: 0199548935. DOI: 10.1093/acprof:oso/9780199548927.001.0001 (see page 8).

[Dou05] Yon Dourisboure. **Compact Routing Schemes for Generalised Chordal Graphs**. *Journal of Graph Algorithms and Applications* 9:2 (2005), 277–297. DOI: 10.7155/jgaa.00109 (see page 50).

[DP12] Devdatt P. Dubhashi and Alessandro Panconesi. **Concentration of Measure for the Analysis of Randomized Algorithms**. Cambridge University Press, 2012. ISBN: 978-1-107-60660-9 (see page 17).

[DYL04] Feodor Dragan, Chenyu Yan, and Irina Lomonosov. **Collective Tree Spanners of Graphs**. In: *SIAM Journal on Discrete Mathematics*. Vol. 20. 2004. DOI: 10.1007/978-3-540-27810-8_7 (see page 50).

[EG08] David Eppstein and Michael T Goodrich. **Succinct Greedy Graph Drawing in the Hyperbolic Plane**. In: *International Symposium on Graph Drawing*. 2008, 14–25. DOI: 10.1007/978-3-642-00219-9_3 (see page 51).

[EG11] David Eppstein and Michael T. Goodrich. **Succinct Greedy Geometric Routing Using Hyperbolic Geometry**. *IEEE Transactions on Computers* 60:11 (2011), 1571–1580. DOI: 10.1109/TC.2010.257 (see page 51).

[EGP03] Tamar Eilam, Cyril Gavoille, and David Peleg. **Compact Routing Schemes with Low Stretch Factor**. *Journal of Algorithms* 46:2 (2003), 97–114. DOI: 10.1016/S0196-6774(03)00002-6 (see pages 50, 51, 57).

[EHT65] Paul Erdös, Frank Harary, and William T. Tutte. **On the Dimension of a Graph**. *Mathematika* 12:2 (1965), 118–122. DOI: 10.1112/S0025579300005222 (see page 168).

[ELW16] Leah Epstein, Asaf Levin, and Gerhard J. Woeginger. **Vertex Cover Meets Scheduling**. *Algorithmica* 74 (2016), 1148–1173. DOI: 10.1007/s00453-015-9992-y (see page 109).

[ER59] Paul Erdős and Alfréd Rényi. **On Random Graphs I**. *Publicationes Mathematicae Debrecen* 6 (1959), 290–297 (see page 3).

[EWG08] M. Enachescu, M. Wang, and A. Goel. **Reducing Maximum Stretch in Compact Routing**. In: *27th IEEE Conference on Computer Communications (INFOCOM)*. 2008, 336–340. DOI: 10.1109/INFOCOM.2008.76 (see page 50).

[FGK09] Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. **A Measure & Conquer Approach for the Analysis of Exact Algorithms**. *Journal of the ACM* 56:5 (2009), 25:1–25:32. DOI: 10.1145/1552285.1552286 (see page 111).

[Fil+07] Eric Filiol, Edouard Franc, Alessandro Gubbioli, Benoit Moquet, and Guillaume Roblot. **Combinatorial Optimisation of Worm Propagation on an Unknown Network**. *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 1 (2007), 2931–2937 (see page 109).

[Fis04] Aleksei V. Fishkin. **Disk Graphs: A Short Survey**. In: *Approximation and Online Algorithms*. 2004, 260–264. DOI: 10.1007/978-3-540-24592-6_23 (see page 6).

[FK15a] Tobias Friedrich and Anton Krohmer. **On the Diameter of Hyperbolic Random Graphs**. In: *42nd International Colloquium on Automata, Languages, and Programming (ICALP)*. 2015, 614–625. DOI: 10.1007/978-3-662-47666-6_49 (see pages 78, 81).

[FK15b] Alan Frieze and Michał Karoński. **Introduction to Random Graphs**. Cambridge University Press, 2015. DOI: 10.1017/CBO9781316339831 (see page 5).

[FK18] Tobias Friedrich and Anton Krohmer. **On the Diameter of Hyperbolic Random Graphs**. *SIAM Journal on Discrete Mathematics* 32:2 (2018), 1314–1334. DOI: 10.1137/17M1123961 (see pages 6, 40–43).

[FKS21] Tobias Friedrich, Maximilian Katzmann, and Leon Schiller. **Computing Voronoi Diagrams in the Polar-Coordinate Model of the Hyperbolic Plane**. *CoRR* (2021). URL: https://arxiv.org/abs/2112.02553 (see page 170).

[FM18] Nikolaos Fountoulakis and Tobias Müller. **Law of Large Numbers for the Largest Component in a Hyperbolic Model of Complex Networks**. *The Annals of Applied Probability* 28:1 (2018), 607–650. DOI: 10.1214/17-AAP1314 (see page 42).

[Fou+21] Nikolaos Fountoulakis, Pim van der Hoorn, Tobias Müller, and Markus Schepers. **Clustering in a Hyperbolic Model of Complex Networks**. *Electronic Journal of Probability* 26:none (2021), 1–132. DOI: 10.1214/21-EJP583 (see pages 41, 42).

[Fou15] Nikolaos Fountoulakis. **On a Geometrization of the Chung–Lu Model For Complex Networks**. *Journal of Complex Networks* 3:3 (2015), 361–387. DOI: 10.1093/comnet/cnu049 (see pages 5, 6).

[FPW09] R. Flury, S. V. Pemmaraju, and R. Wattenhofer. **Greedy Routing with Bounded Stretch**. In: *28th IEEE Conference on Computer Communications (INFOCOM)*. 2009, 1737–1745. DOI: 10.1109/INFCOM.2009.5062093 (see pages 48, 51, 52, 57).

[Fre+17] Ofer Freedman, Paweł Gawrychowski, Patrick K. Nicholson, and Oren Weimann. **Optimal Distance Labeling Schemes for Trees**. In: *Proceedings of the ACM Symposium on Principles of Distributed Computing*. 2017, 185–194. DOI: 10.1145/3087801.3087804 (see pages 48, 58).

[FSS13] Tobias Friedrich, Thomas Sauerwald, and Alexandre Stauffer. **Diameter and Broadcast Time of Random Geometric Graphs in Arbitrary Dimensions**. *Algorithmica* (2013), 65–88. DOI: 10.1007/s00453-012-9710-y (see page 75).

[FY21] Max Franck and Sorrachai Yingchareonthawornchai. **Engineering Nearly Linear-Time Algorithms for Small Vertex Connectivity**. In: *19th International Symposium on Experimental Algorithms (SEA)*. 2021, 1:1–1:18. DOI: 10.4230/LIPIcs.SEA.2021.1 (see page 167).

[Gan66] David Gans. **A New Model of the Hyperbolic Plane**. *The American Mathematical Monthly* 73:3 (1966), 291–295 (see page 23).

[Gar+16] Guillermo García-Pérez, Marián Boguñá, Antoine Allard, and M. Serrano. **The Hidden Hyperbolic Geometry of International Trade: World Trade Atlas 1870–2013**. *Scientific Reports* 6 (2016), 33441. DOI: 10.1038/srep33441 (see page 6).

[Gav+04] Cyril Gavoille, David Peleg, Stéphane Pérennes, and Ran Raz. **Distance Labeling in Graphs**. *Journal of Algorithms* 53:1 (2004), 85–112. DOI: 10.1016/j.jalgor.2004.05.002 (see pages 55, 57).

[GH14] Mikael Gast and Mathias Hauptmann. **Approximability of the Vertex Cover Problem In Power-Law Graphs**. *Theoretical Computer Science* 516 (2014), 60–70. DOI: 10.1016/j.tcs.2013.11.004 (see page 134).

[Gil59] Edgar Nelson Gilbert. **Random Graphs**. *The Annals of Mathematical Statistics* 30:4 (1959), 1141–1144. ISSN: 00034851. URL: http://www.jstor.org/stable/2237458 (see page 3).

[GL05] Cyril Gavoille and Olivier Ly. **Distance Labeling in Hyperbolic Graphs**. In: *Algorithms and Computation*. Vol. 3827. 2005, 1071–1079. DOI: 10.1007/11602613_106 (see page 51).

[GM05] Alison P. Galvani and Robert M. May. **Dimensions of Superspreading**. *Nature* 438 (2005), 293–295. DOI: 10.1038/438293a (see page 169).

[GN72] Robert Garfinkel and George L. Nemhauser. **Integer Programming**. John Wiley & Sons, 1972 (see page 111).

[GP96] Cyril Gavoille and Stéphane Pérennès. **Memory Requirement for Routing in Distributed Networks**. In: *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*. 1996, 125–133. DOI: 10.1145/248052.248075 (see page 50).

[GPP12] Luca Gugelmann, Konstantinos Panagiotou, and Ueli Peter. **Random Hyperbolic Graphs: Degree Sequence and Clustering**. In: *39th International Colloquium on Automata, Languages, and Programming (ICALP)*. 2012, 573–585. DOI: 10.1007/978-3-642-31585-5_51 (see pages 6, 39, 40, 42, 44, 81, 85, 107).

[GS09] Michael T. Goodrich and Darren Strash. **Succinct Greedy Geometric Routing in the Euclidean Plane**. In: *Algorithms and Computation*. 2009, 781–791 (see page 51).

[GV21] Vijay Ganesh and Moshe Y. Vardi. "On the Unreasonable Effectiveness of SAT Solvers." In: *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021, 547–566. DOI: 10.1017/9781108637435.032 (see page 2).

[Hof16] Remco van der Hofstad. **Random Graphs and Complex Networks**. Vol. 1. Cambridge University Press, 2016. DOI: 10.1017/9781316779422 (see page 5).

[Hou+14] R. Houthooft, S. Sahhaf, W. Tavernier, F. De Turck, D. Colle, and M. Pickavet. **Fault-tolerant Greedy Forest Routing for Complex Networks**. In: *2014 6th International Workshop on Reliable Networks Design and Modeling (RNDM)*. 2014, 1–8. DOI: 10.1109/RNDM.2014.7014924 (see pages 48, 57).

[HWR16] Matthew Hall, Mark Woolhouse, and Andrew Rambaut. **Epidemic Reconstruction in a Phylogenetics Framework: Transmission Trees as Partitions of the Node Set**. *PLOS Computational Biology* 11:12 (2016), 1–36. DOI: 10.1371/journal.pcbi.1004613 (see page 170).

[Joh74] David S. Johnson. **Approximation Algorithms for Combinatorial Problems**. *Journal of Computer and System Sciences* 9:3 (1974), 256–278. DOI: 10.1016/S0022-0000(74)80044-9 (see page 133).

[Kar09] George Karakostas. **A Better Approximation Ratio for the Vertex Cover Problem**. *ACM Transactions on Algorithms* 5:4 (2009), 41:1–41:8. DOI: 10.1145/1597036.1597045 (see page 133).

[Kar72] Richard M. Karp. **Reducibility Among Combinatorial Problems**. In: *Proceedings of a Symposium on the Complexity of Computer Computations*. 1972, 85–103 (see pages 1, 9, 109).

[KE02] Konstantin Klemm and Víctor M. Eguíluz. **Growing Scale-free Networks with Small-World Behavior**. *Physical Review E* 65 (2002), 057102. DOI: 10.1103/PhysRevE.65.057102 (see page 5).

[Kel+21] Leon Kellerhals, Tomohiro Koana, André Nichterlein, and Philipp Zschoche. **The PACE 2021 Parameterized Algorithms and Computational Experiments Challenge: Cluster Editing**. In: *16th International Symposium on Parameterized and Exact Computation (IPEC)*. Vol. 214. 2021, 26:1–26:18. DOI: 10.4230/LIPIcs.IPEC.2021.26 (see page 2).

[Keu18] Ralph Keusch. **Geometric Inhomogeneous Random Graphs and Graph Coloring Games**. PhD thesis. ETH Zurich, 2018. DOI: 10.3929/ethz-b-000269658 (see pages 40, 81, 82, 134).

[KFY04] Dmitri Krioukov, Kevin Fall, and Xiaowei Yang. **Compact Routing on Internet-like Graphs**. In: *IEEE INFOCOM 2004*. Vol. 1. 2004, 219. DOI: 10.1109/INFCOM.2004.1354495 (see page 52).

[Kis20] Sándor Kisfaludi-Bak. **Hyperbolic Intersection Graphs and (Quasi)-Polynomial Time**. In: *Symposium on Discrete Algorithms (SODA)*. 2020, 1621–1638. DOI: 10.1137/1.9781611975994.100 (see pages 38, 167).

[KK00] Brad Karp and Hsiang-Tsung Kung. **GPSR: Greedy Perimeter Stateless Routing for Wireless Networks**. In: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. 2000, 243–254. DOI: 10.1145/345910.345953 (see page 51).

[KK15] Zoltán Király and Sándor Kisfaludi-Bak. **A Succinct Tree Coding for Greedy Navigation**. Tech. rep. TR-2015-02. `egres.elte.hu`. Egerváry Research Group, Budapest, 2015 (see page 51).

[KL07] Robert Krauthgamer and James R. Lee. **The Intrinsic Dimensionality of Graphs**. *Combinatorica* 27:2 (2007), 551–585. DOI: 10.1007/s00493-007-2183-y (see page 168).

[KL20] Júlia Komjáthy and Bas Lodewijks. **Explosion in Weighted Hyperbolic Random Graphs And Geometric Inhomogeneous Random Graphs**. *Stochastic Processes and their Applications* 130:3 (2020), 1309–1367. DOI: 10.1016/j.spa.2019.04.014 (see page 168).

[Kle00] Jon Kleinberg. **The Small-World Phenomenon: An Algorithmic Perspective**. In: *Thirty-Second Annual ACM Symposium on Theory of Computing (STOC)*. 2000, 163–170. DOI: 10.1145/335305.335325 (see page 52).

[Kle07] Robert Kleinberg. **Geographic Routing Using Hyperbolic Space**. In: *26th IEEE International Conference on Computer Communications (INFOCOM)*. 2007, 1902–1909. DOI: 10.1109/INFCOM.2007.221 (see page 51).

[KM15] Marcos A. Kiwi and Dieter Mitsche. **A Bound for the Diameter of Random Hyperbolic Graphs**. In: *Twelfth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*. 2015, 26–39. DOI: 10.1137/1.9781611973761.3 (see pages 6, 42, 43).

[KM18] Marcos Kiwi and Dieter Mitsche. **Spectral Gap of Random Hyperbolic Graphs And Related Parameters**. *The Annals of Applied Probability* 28:2 (2018), 941–989. DOI: 10.1214/17-AAP1323 (see page 41).

[KM19] Marcos Kiwi and Dieter Mitsche. **On the Second Largest Component of Random Hyperbolic Graphs**. *SIAM Journal on Discrete Mathematics* 33:4 (2019), 2200–2217. DOI: 10.1137/18M121201X (see page 41).

[KR08] Subhash Khot and Oded Regev. **Vertex Cover Might Be Hard to Approximate to Within $2-\varepsilon$**. *Journal of Computer and System Sciences* 74:3 (2008), 335–349. DOI: 10.1016/j.jcss.2007.06.019 (see page 133).

[Kri+07] Dmitri Krioukov, k c claffy k c, Kevin Fall, and Arthur Brady. **On Compact Routing for the Internet**. *SIGCOMM Comput. Commun. Rev.* 37:3 (2007), 41–52. DOI: 10.1145/1273445.1273450 (see page 52).

[Kri+10] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. **Hyperbolic Geometry of Complex Networks**. *Physical Review E* 82 (2010), 036106. DOI: 10.1103/PhysRevE.82.036106 (see pages 6, 23, 39, 167).

[Kro16] Anton Krohmer. **Structures & Algorithms in Hyperbolic Random Graphs**. PhD thesis. University of Potsdam, 2016 (see pages 21, 39, 44).

[Kun13] Jérôme Kunegis. **KONECT: The Koblenz Network Collection**. In: *International Conference on World Wide Web (WWW)*. 2013, 1343–1350. DOI: 10.1145/2487788.2488173 (see pages 129, 161).

[LFC17] Zheng Lu, Yunhe Feng, and Qing Cao. **Decentralized Search for Shortest Path Approximation in Large-Scale Complex Networks**. In: *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. 2017, 130–137. DOI: 10.1109/CloudCom.2017.36 (see page 71).

[LK14] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data. 2014 (see page 129).

[LR89] Michael Luby and Prabhakar Ragde. **A Bidirectional Shortest-path Algorithm with Good Average-Case Behavior**. *Algorithmica* 4:1 (1989), 551–567. DOI: 10.1007/BF01553908 (see page 72).

[MPP07] Frosso S. Makri, Andreas N. Philippou, and Zaharias M. Psillakis. **Success Run Statistics Defined on an Urn Model**. *Advances in Applied Probability* 39:4 (2007), 991–1019. DOI: 10.1239/aap/1198177236 (see page 144).

[MS19] Tobias Müller and Merlijn Staps. **The Diameter of KPKVB Random Graphs**. *Advances in Applied Probability* 51:2 (2019), 358–377. DOI: 10.1017/apr.2019.23 (see pages 6, 41, 67).

[MU05] Michael Mitzenmacher and Eli Upfal. **Probability and Computing: Randomized Algorithms and Probabilistic Analysis**. Cambridge University Press, 2005. DOI: 10.1017/CBO9780511813603 (see page 15).

[Muh07] Rashid Bin Muhammad. **A Distributed Geometric Routing Algorithm for Ad HocWireless Networks**. In: *Fourth International Conference on Information Technology (ITNG)*. 2007, 961–963. DOI: 10.1109/ITNG.2007.4 (see page 51).

[Mus+17] Alessandro Muscoloni, Josephine Maria Thomas, Sara Ciucci, Ginestra Bianconi, and Carlo Vittorio Cannistraci. **Machine Learning Meets Complex Networks Via Coalescent Embedding in the Hyperbolic Space**. *Nature Communications* 8:1 (2017), 1615. DOI: 10.1038/s41467-017-01825-5 (see page 169).

[MZ21] Claire Mathieu and Hang Zhou. **A Simple Algorithm for Graph Reconstruction**. In: *29th Annual European Symposium on Algorithms (ESA)*. 2021, 68:1–68:18. DOI: 10.4230/LIPIcs.ESA.2021.68 (see page 167).

[New01] Mark Newman. **Clustering and Preferential Attachment in Growing Networks**. *Physical Review E* 64 (2001), 025102. DOI: 10.1103/PhysRevE.64.025102 (see page 5).

[PAK15] Fragkiskos Papadopoulos, Rodrigo Aldecoa, and Dmitri Krioukov. **Network Geometry Inference Using Common Neighbors**. *Physical Review E* 92:2 (2015), 022807. DOI: 10.1103/PhysRevE.92.022807 (see page 169).

[Pap+10] Fragkiskos Papadopoulos, Dmitri Krioukov, Marián Boguñá, and Amin Vahdat. **Greedy Forwarding in Dynamic Scale-Free Networks Embedded in Hyperbolic Metric Spaces**. In: *2010 Proceedings IEEE INFOCOM*. 2010, 1–9. DOI: 10.1109/INFCOM.2010.5462131 (see page 52).

[Pen+12] Wei Peng, Xiaofeng Hu, Feng Zhao, and Jinshu Su. **A Fast Algorithm to Find All-Pairs Shortest Paths in Complex Networks**. *Procedia Computer Science* 9 (2012), 557–566. DOI: 10.1016/j.procs.2012.04.060 (see page 71).

[Pen+21] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. **Hyperbolic Deep Neural Networks: A Survey**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. DOI: 10.1109/TPAMI.2021.3136921 (see pages 168, 169).

[Pen03] Mathew Penrose. **Random Geometric Graphs**. Oxford University Press, 2003. ISBN: 9780198506263 (see pages 5, 75).

[Poh69] Ira Sheldon Pohl. **Bi-directional and Heuristic Search in Path Problems**. PhD thesis. Stanford University, 1969 (see page 71).

[PPK15] Fragkiskos Papadopoulos, Constantinos Psomas, and Dmitri Krioukov. **Network Mapping by Replaying Hyperbolic Growth**. *IEEE/ACM Transactions on Networking* 23:1 (2015), 198–211. DOI: 10.1109/TNET.2013.2294052 (see page 169).

[PR05] Christos Papadimitriou and David Ratajczak. **On a Conjecture Related to Geometric Routing**. *Theoretical Computer Science* 344:1 (2005), 3–14. DOI: 10.1016/j.tcs.2005.06.022 (see page 51).

[PR14] Mihai Pătraşcu and Liam Roditty. **Distance Oracles beyond the Thorup–Zwick Bound**. *SIAM Journal on Computing* 43:1 (2014), 300–311. DOI: 10.1137/11084128X (see page 50).

[PRT12] Mihai Pătraşcu, Liam Roditty, and Mikkel Thorup. **A New Infinity of Distance Oracles for Sparse Graphs**. In: *IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*. 2012, 738–747. DOI: 10.1109/FOCS.2012.44 (see pages 50, 52).

[PS89] David Peleg and Alejandro A. Schäffer. **Graph spanners**. *Journal of Graph Theory* 13:1 (1989), 99–116. DOI: 10.1002/jgt.3190130114 (see page 54).

[PU89] David Peleg and Eli Upfal. **A Trade-off between Space and Efficiency for Routing Tables**. *Journal of the ACM* 36:3 (1989), 510–530. DOI: 10.1145/65950.65953 (see pages 48, 50, 51).

[PW05] Kihong Park and Walter Williger. **The Internet As a Large-Scale Complex System**. Oxford University Press, Inc., 2005. ISBN: 0195157214 (see page 133).

[RA15] Ryan A. Rossi and Nesreen K. Ahmed. **The Network Data Repository with Interactive Graph Analytics and Visualization**. In: *AAAI*. 2015. URL: http://networkrepository.com (see pages 68, 129, 161).

[Rao+03] Ananth Rao, Sylvia Ratnasamy, Christos Papadimitriou, Scott Shenker, and Ion Stoica. **Geographic Routing without Location Information**. In: *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*. 2003, 96–108. DOI: 10.1145/938985.938996 (see page 51).

[Ros19] Sheldon M. Ross. **A First Course in Probability**. Pearson Education Limited, 2019 (see page 12).

[RR95] Arlan Ramsay and Robert D. Richtmyer. **Introduction to Hyperbolic Geometry**. Springer, 1995. ISBN: 978-1-4757-5585-5. DOI: 10.1007/978-1-4757-5585-5 (see page 23).

[RRŠ89] Jan Reiterman, Vojtěch Rödl, and Edita Šiňajová. **Geometrical Embeddings of Graphs**. *Discrete Mathematics* 74:3 (1989), 291–319. DOI: 10.1016/0012-365X(89)90142-8 (see page 168).

[RT16] Liam Roditty and Roei Tov. **Close to Linear Space Routing Schemes**. *Distributed Computing* 29:1 (2016), 65–74. DOI: 10.1007/s00446-015-0256-5 (see page 50).

[Ser+21] Matteo Serafino, Giulio Cimini, Amos Maritan, Andrea Rinaldo, Samir Suweis, Jayanth R. Banavar, and Guido Caldarelli. **True Scale-free Networks Hidden by Finite Size Effects**. *Proceedings of the National Academy of Sciences* 118:2 (2021). DOI: 10.1073/pnas.2013825118 (see page 4).

[SGS13] Mariana O. da Silva, Gustavo A. Gimenez-Lugo, and Murilo V. G. da Silva. **Vertex Cover in Complex Networks**. *International Journal of Modern Physics C* 24:11 (2013), 1350078. DOI: 10.1142/S0129183113500782 (see pages 134, 161).

[SH75] Michael Ian Shamos and Dan Hoey. **Closest-point Problems**. In: *16th Annual Symposium on Foundations of Computer Science (SFCS)*. 1975, 151–162. DOI: 10.1109/SFCS.1975.8 (see page 170).

[SK85] Nicola Santoro and Ramez Khatib. **Labelling and Implicit Routing in Networks**. *The Computer Journal* 28:1 (1985), 5–8. DOI: 10.1093/comjnl/28.1.5 (see page 51).

[SMS18] Khot Subhash, Dor Minzer, and Muli Safra. **Pseudorandom Sets in Grassmann Graph Have Near-Perfect Expansion**. In: *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. 2018, 592–601. DOI: 10.1109/FOCS.2018.00062 (see page 133).

[ST09] Daniel A. Spielman and Shang-Hua Teng. **Smoothed Analysis: An Attempt to Explain the Behavior of Algorithms in Practice**. *Communications of the ACM* 52:10 (2009), 76–84. DOI: 10.1145/1562764.1562785 (see page 2).

[Ste20] Clara Stegehuis. **Closure Coefficients in Scale-free Complex Networks**. *Journal of Complex Networks* 8:3 (2020). DOI: 10.1093/comnet/cnaa020 (see page 41).

[Tam+17] Hisao Tamaki, Hiromu Ohtsuka, Takuto Sato, and Keitaro Makii. *TCS-Meiji PACE2017-TrackA*. github.com/TCS-Meiji/PACE2017-TrackA. 2017 (see page 129).

[Tan+10] Mingdong Tang, Hongyang Chen, Guoqing Zhang, and Jing Yang. **Tree Cover Based Geographic Routing with Guaranteed Delivery**. In: *IEEE International Conference on Communications*. 2010, 1–5. DOI: 10.1109/ICC.2010.5502391 (see page 57).

[Tho04] Mikkel Thorup. **Compact Oracles for Reachability and Approximate Distances in Planar Digraphs**. *Journal of the ACM* 51:6 (2004), 993–1024. DOI: 10.1145/1039488.1039493 (see page 50).

[TK84] Hideaki Takagi and Leonard Kleinrock. **Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals**. *IEEE Transactions on Communications* 32:3 (1984), 246–257. DOI: 10.1109/TCOM.1984.1096061 (see page 51).

[TM69] Jeffrey Travers and Stanley Milgram. **An Experimental Study of the Small World Problem**. *Sociometry* 32:4 (1969), 425–443. URL: http://www.jstor.org/stable/2786545 (see page 6).

[TYZ09] Mingdong Tang, Jing Yang, and Guoqing Zhang. **Compact Routing on Random Power Law Graphs**. In: *8th IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC)*. 2009, 575–578. DOI: 10.1109/DASC.2009.133 (see pages 50, 52).

[TZ01] Mikkel Thorup and Uri Zwick. **Compact Routing Schemes**. In: *Proceedings of the Thirteenth Annual ACM Symposium on Parallel Algorithms and Architectures*. 2001, 1–10. DOI: 10.1145/378580.378581 (see pages 50, 52, 57, 58).

[TZ05] Mikkel Thorup and Uri Zwick. **Approximate Distance Oracles**. *Journal of the ACM* 52:1 (2005), 1–24. DOI: 10.1145/1044731.1044732 (see page 50).

[Voi+19] Ivan Voitalov, Pim van der Hoorn, Remco van der Hofstad, and Dmitri Krioukov. **Scale-free Networks Well Done**. *Physical Review Research* 1 (3 2019), 033034. DOI: 10.1103/PhysRevResearch.1.033034 (see page 4).

[VS14] Kevin Verbeek and Subhash Suri. **Metric Embedding, Hyperbolic Space, and Social Networks**. In: *Thirtieth Annual Symposium on Computational Geometry (SoCG)*. 2014, 501–510. DOI: 10.1145/2582112.2582139 (see page 6).

[VS16] André L. Vignatti and Murilo V.G. da Silva. **Minimum Vertex Cover in Generalized Random Graphs with Power Law Degree Distribution**. *Theoretical Computer Science* 647 (2016), 101–111. DOI: 10.1016/j.tcs.2016.08.002 (see page 134).

[War16] Lutz Warnke. **On the Method of Typical Bounded Differences**. *Combinatorics, Probability and Computing* 25:2 (2016), 269–299. DOI: 10.1017/S0963548315000103 (see page 18).

[WP09] Cédric Westphal and Guanhong Pei. **Scalable Routing Via Greedy Embedding**. In: *IEEE INFOCOM 2009*. 2009, 2826–2830. DOI: 10.1109/INFCOM.2009.5062240 (see page 51).

[WS98] Duncan J. Watts and Steven H. Strogatz. **Collective Dynamics of 'Small-world' Networks**. *Nature* 393 (1998), 440–442. DOI: 10.1038/30918 (see pages 5, 8).

[XN17] Mingyu Xiao and Hiroshi Nagamochi. **Exact Algorithms for Maximum Independent Set**. *Information and Computation* 255 (2017), 126–146. DOI: 10.1016/j.ic.2017.06.001 (see pages 109, 159).

[YR20] Weihua Yang and David Rideout. **High Dimensional Hyperbolic Geometry of Complex Networks**. *Mathematics* 8:11 (2020). DOI: 10.3390/math8111861 (see page 168).

[ZG11] Huaming Zhang and Swetha Govindaiah. **Greedy Routing via Embedding Graphs onto Semi-metric Spaces**. In: *Frontiers in Algorithmics and Algorithmic Aspects in Information and Management*. 2011, 58–69 (see page 51).

[ZG13] Huaming Zhang and Swetha Govindaiah. **Greedy Routing via Embedding Graphs onto Semi-Metric Spaces**. *Theoretical Computer Science* 508 (2013), 26–34. DOI: 10.1016/j.tcs.2012.01.049 (see page 53).

List of Publications

Articles in Refereed Journals

[1] **Solving Vertex Cover in Polynomial Time on Hyperbolic Random Graphs**. *Theory of Computing Systems* (2021). DOI: 10.1007/s00224-021-10062-9. Joint work with Thomas Bläsius, Philipp Fischbeck, and Tobias Friedrich.

[2] **Hyperbolic Embeddings for Near-Optimal Greedy Routing**. *ACM Journal of Experimental Algorithmics* 25 (2020). DOI: 10.1145/3381751. Joint work with Thomas Bläsius, Tobias Friedrich, and Anton Krohmer.

[3] **Efficient Shortest Paths in Scale-Free Networks with Underlying Hyperbolic Geometry**. *ACM Trans. Algorithms* 18 (2022), 19:1–19:32. DOI: 10.1145/3516483. Joint work with Thomas Bläsius, Cedric Freiberger, Tobias Friedrich, Felix Montenegro-Retana, and Marianne Thieffry.

[4] **Unbounded Discrepancy of Deterministic Random Walks on Grids**. *SIAM Journal on Discrete Mathematics* 32:4 (2018), 2441–2452. Joint work with Tobias Friedrich and Anton Krohmer.

Articles in Refereed Conference Proceedings

[5] **Towards explainable real estate valuation via evolutionary algorithms**. In: *Genetic and Evolutionary Computation Conference GECCO*. 2022, 1130–1138. DOI: 10.1145/3512290.3528801. Joint work with Sebastian Angrick, Ben Bals, Niko Hastrich, Maximilian Kleissl, Jonas Schmidt, Vanja Doskoc, Louise Molitor, and Tobias Friedrich.

[6] **Efficiently Approximating Vertex Cover on Scale-Free Networks with Underlying Hyperbolic Geometry**. In: *29th Annual European Symposium on Algorithms (ESA)*. 2021, 20:1–20:15. DOI: 10.4230/LIPIcs.ESA.2021.20. Joint work with Thomas Bläsius and Tobias Friedrich.

[7] **Force-Directed Embedding of Scale-Free Networks in the Hyperbolic Plane**. In: *19th International Symposium on Experimental Algorithms (SEA)*. 2021, 22:1–22:18. DOI: 10.4230/LIPIcs.SEA.2021.22. Joint work with Thomas Bläsius and Tobias Friedrich.

[8] **Solving Vertex Cover in Polynomial Time on Hyperbolic Random Graphs**. In: *37th International Symposium on Theoretical Aspects of Computer Science (STACS)*. 2020, 25:1–25:14. DOI: 10.4230/LIPIcs.STACS.2020.25. Joint work with Thomas Bläsius, Philipp Fischbeck, and Tobias Friedrich.

[9] **Hyperbolic Embeddings for Near-Optimal Greedy Routing**. In: *Algorithm Engineering and Experiments (ALENEX)*. 2018, 199–208. DOI: 10.1137/1.9781611975055.17. Joint work with Thomas Bläsius, Tobias Friedrich, and Anton Krohmer.

[10] **Strongly Hyperbolic Unit Disk Graphs**. In: *40th International Symposium on Theoretical Aspects of Computer Science (STACS)*. Accepted for publication. 2023. Joint work with Thomas Bläsius, Tobias Friedrich, and Daniel Stephan.

[11] **Efficient Shortest Paths in Scale-Free Networks with Underlying Hyperbolic Geometry**. In: *45th International Colloquium on Automata, Languages, and Programming (ICALP)*. 2018, 20:1–20:14. DOI: 10.4230/LIPIcs.ICALP.2018.20. Joint work with Thomas Bläsius, Cedric Freiberger, Tobias Friedrich, Felix Montenegro-Retana, and Marianne Thieffry.

[12] **Towards a Systematic Evaluation of Generative Network Models**. In: *Workshop on Algorithms and Models for the Web Graph (WAW)*. 2018, 99–114. Joint work with Thomas Bläsius, Tobias Friedrich, Anton Krohmer, and Jonathan Striebel.

[13] **Efficiently Generating Geometric Inhomogeneous and Hyperbolic Random Graphs**. In: *27th Annual European Symposium on Algorithms (ESA 2019)*. 2019, 21:1–21:14. DOI: 10.4230/LIPIcs.ESA.2019.21. Joint work with Thomas Bläsius, Tobias Friedrich, Ulrich Meyer, Manuel Penschuck, and Christopher Weyand.

[14] **Unbounded Discrepancy of Deterministic Random Walks on Grids**. In: *International Symposium on Algorithms and Computation (ISAAC)*. 2015, 212–222. Joint work with Tobias Friedrich and Anton Krohmer.

[15] **Algorithms for Hard-Constraint Point Processes via Discretization**. In: *Computing and Combinatorics - 28th International Conference, COCOON*. 2022, 242–254. DOI: 10.1007/978-3-031-22105-7_22. Joint work with Tobias Friedrich, Andreas Göbel, Martin S. Krejca, and Marcus Pappik.

[16] **Systematic Exploration of Larger Local Search Neighborhoods for the Minimum Vertex Cover Problem**. In: *Conference on Artificial Intelligence (AAAI)*. 2017, 846–852. Joint work with Christian Komusiewicz.