

# Efficiently Approximating Vertex Cover on Scale-Free Networks with Underlying Hyperbolic Geometry

Thomas Bläsius ✉

Karlsruhe Institute of Technology, Germany

Tobias Friedrich ✉ 

Hasso Plattner Institute, University of Potsdam, Germany

Maximilian Katzmann ✉ 

Hasso Plattner Institute, University of Potsdam, Germany

---

## Abstract

---

Finding a minimum vertex cover in a network is a fundamental NP-complete graph problem. One way to deal with its computational hardness, is to trade the qualitative performance of an algorithm (allowing non-optimal outputs) for an improved running time. For the vertex cover problem, there is a gap between theory and practice when it comes to understanding this tradeoff. 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.

A promising approach towards understanding this discrepancy is to recognize the differences between theoretical worst-case instances and real-world networks. Following this direction, we close the gap between theory and practice by providing an algorithm that efficiently computes nearly optimal vertex cover approximations on hyperbolic random graphs; a network model that closely resembles real-world networks in terms of degree distribution, clustering, and the small-world property. More precisely, our algorithm computes a  $(1 + o(1))$ -approximation, asymptotically almost surely, and has a running time of  $\mathcal{O}(m \log(n))$ .

The proposed algorithm is an adaption 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 tradeoff 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.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Approximation algorithms analysis; Theory of computation  $\rightarrow$  Random network models; Mathematics of computing  $\rightarrow$  Approximation algorithms; Mathematics of computing  $\rightarrow$  Random graphs

**Keywords and phrases** vertex cover, approximation, random graphs, hyperbolic geometry, efficient algorithm

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2021.20

**Related Version** *Full Version:* <https://arxiv.org/abs/2010.02787> [7]

**Funding** This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Grant No. 390859508.

## 1 Introduction

A *vertex cover* of a graph is a subset of the vertices that leaves the graph edgeless upon deletion. Since the problem of finding a smallest vertex cover is NP-complete [19], there are probably no algorithms that solve it efficiently. Nevertheless, the problem is highly relevant due to its applications in computational biology [1], scheduling [13], and internet security [14]. Therefore, there is an ongoing effort in exploring methods that can be used in practice [2, 3], and while they often work well, they still cannot guarantee efficient running times.



© Thomas Bläsius, Tobias Friedrich, and Maximilian Katzmann; licensed under Creative Commons License CC-BY 4.0

29th Annual European Symposium on Algorithms (ESA 2021).

Editors: Petra Mutzel, Rasmus Pagh, and Grzegorz Herman; Article No. 20; pp. 20:1–20:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 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 guarantee a better approximation ratio than  $\log(n)$ , i.e., there are graphs where it produces a vertex cover whose size exceeds the one of an optimum by a factor of  $\log(n)$ . This can be improved to a 2-approximation using a simple linear-time algorithm. The best known polynomial time approximation reduces the factor to  $2 - \Theta(\log(n)^{-1/2})$  [18]. 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  $\varepsilon > 0$  [20] and it is proven that finding a  $\sqrt{2}$ -approximation is NP-hard [26].

Therefore, it is very surprising that the standard greedy algorithm not only beats the 2-approximation on autonomous systems graphs like the internet [25], it also performs well on many real-world networks, obtaining approximation ratios that are very close to 1 [11]. So the theoretical bounds do not match what is observed in practice. One approach to explaining this discrepancy is to consider the differences between the examined instances. Theoretical bounds are often obtained by designing worst-case instances. However, real-world networks rarely resemble the worst case. More realistic statements can be obtained by making assumptions about the solution space [4, 9], or by restricting the analysis to networks with properties that are observed in the real world.

Many real networks, like social networks, communication networks, or protein-interaction networks, are considered to be *scale-free*. Such graphs feature a power-law degree distribution (only few vertices have high degree, while many vertices have low degree), high clustering (two vertices are likely to be adjacent if they have a common neighbor), and a small diameter.

Previous efforts to obtain more realistic 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 [10, 16, 28]. 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 [10]. Moreover, it was shown that it is possible to compute an expected  $(2 - \varepsilon)$ -approximation for a constant  $\varepsilon$ , in polynomial time on such networks [16] and this was later improved to about 1.7 depending on properties of the distribution [28]. However, it was also shown that even on graphs that have a power-law degree distribution, the vertex cover problem remains NP-hard to approximate within some constant factor [10]. 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 paper is to close this gap between theory and practice, by considering a random graph model that features all of the three mentioned properties of scale-free networks. The *hyperbolic random graph model* was introduced by Krioukov et al. [21] and it was shown that the graphs generated by the model have a power-law degree distribution and high clustering [17], as well as a small diameter [24]. Consequently, they are good representations of many real-world networks [8, 15, 27]. Additionally, the model is conceptually simple, making it accessible to mathematical analysis. Therefore, it has proven to be a useful framework to theoretically explain why algorithms work well in practice [6]. In fact, it has been shown that the vertex cover problem can be solved exactly in polynomial time on hyperbolic random graphs, with high probability [5]. However, we note that 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.

In this paper, 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 hyperbolic random graphs, this algorithm achieves an approximation ratio of  $1 + o(1)$ , asymptotically almost surely, and maintains an efficient running time of  $\mathcal{O}(m \log(n))$ , where  $n$  and  $m$  denote the number of vertices and edges in the graph, respectively. Since the average degree of hyperbolic random graphs is constant [17], this implies a quasi-linear running time on such networks. Moreover, we introduce a parameter that can be used to tune the tradeoff 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%. The evaluation of our experiments can be found in the full version of the paper [7].

## 2 Preliminaries

Let  $G = (V, E)$  be an undirected graph. We denote the number of vertices and edges in  $G$  with  $n$  and  $m$ , respectively. The number of vertices in a set  $S \subseteq V$  is denoted by  $|S|$ . The *neighborhood* of a vertex  $v$  is defined as  $N(v) = \{w \in V \mid \{v, w\} \in E\}$ . The size of the neighborhood, called the *degree* of  $v$ , is denoted by  $\deg(v) = |N(v)|$ . For a subset  $S \subseteq V$ , we use  $G[S]$  to denote the *induced subgraph* of  $G$  obtained by removing all vertices in  $V \setminus S$ .

**The Hyperbolic Plane.** After choosing a designated origin  $O$  in the two-dimensional hyperbolic plane, together with a reference ray starting at  $O$ , a point  $p$  is uniquely identified by its *radius*  $r(p)$ , denoting the hyperbolic distance to  $O$ , and its *angle* (or *angular coordinate*)  $\varphi(p)$ , denoting the angular distance between the reference ray and the line through  $p$  and  $O$ . The hyperbolic distance between two points  $p$  and  $q$  is given by

$$\text{dist}(p, q) = \text{acosh}(\cosh(r(p)) \cosh(r(q)) - \sinh(r(p)) \sinh(r(q)) \cos(\Delta_\varphi(p, q))),$$

where  $\cosh(x) = (e^x + e^{-x})/2$ ,  $\sinh(x) = (e^x - e^{-x})/2$  (both growing as  $e^x/2 \pm o(1)$ ), and  $\Delta_\varphi(p, q) = \pi - |\pi - |\varphi(p) - \varphi(q)||$  denotes the angular distance between  $p$  and  $q$ . If not stated otherwise, we assume that computations on angles are performed modulo  $2\pi$ .

In the hyperbolic plane a disk of radius  $r$  has an area of  $2\pi(\cosh(r) - 1)$  and circumference  $2\pi \sinh(r)$ . Thus, the area and the circumference of such a disk grow exponentially with its radius. In contrast, this growth is polynomial in Euclidean space.

**Hyperbolic Random Graphs.** Hyperbolic random graphs are obtained by distributing  $n$  points independently and uniformly at random within a disk of radius  $R$  and connecting any two of them if and only if their hyperbolic distance is at most  $R$ . See Figure 1 (left) for an example. The disk radius  $R$  (which matches the connection threshold) is given by  $R = 2 \log(n) + C$ , where the constant  $C \in \mathbb{R}$  depends on the average degree of the network, as well as the power-law exponent  $\beta = 2\alpha + 1$  (for  $\alpha \in (1/2, 1)$ ). The coordinates of the vertices are drawn as follows. For vertex  $v$  the angular coordinate, denoted by  $\varphi(v)$ , is drawn uniformly at random from  $[0, 2\pi)$  and the radius of  $v$ , denoted by  $r(v)$ , is sampled according

## 20:4 Approximating Vertex Cover on Networks with Underlying Hyperbolic Geometry

to the probability density function  $\alpha \sinh(\alpha r)/(\cosh(\alpha R) - 1)$  for  $r \in [0, R]$ . Thus,

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

is their joint distribution function for  $r \in [0, R]$ . For  $r > R$ ,  $f(r, \varphi) = 0$ .

We denote areas in the hyperbolic disk with calligraphic capital letters. The set of vertices in an area  $\mathcal{A}$  is denoted by  $V(\mathcal{A})$ . The probability for a given vertex to lie in  $\mathcal{A}$  is given by its measure  $\mu(\mathcal{A}) = \int_{\mathcal{A}} f(r, \varphi) d\varphi dr$ . The hyperbolic distance between two vertices  $u$  and  $v$  increases with increasing angular distance between them. The maximum angular distance such that they are still connected by an edge is bounded by [17, Lemma 6]

$$\theta(r(u), r(v)) = 2e^{(R-r(u)-r(v))/2}(1 + \Theta(e^{R-r(u)-r(v)})). \quad (2)$$

**Hyperbolic Random Graphs with an Expected Number of Vertices.** While the positions of the vertices in a hyperbolic random graph are sampled independently of each other, stochastic dependencies are introduced once the positions of some vertices are known. For example, if all vertices lie in an area  $\mathcal{A}$ , the probability for a vertex to lie outside of  $\mathcal{A}$  is 0. When these dependencies are hard to deal with (which we mention explicitly), we resort to a slightly different hyperbolic random graph model, where the positions are sampled using an inhomogeneous Poisson point process. The result of this process is a *hyperbolic random graph with  $n$  vertices in expectation*. There, the number of vertices in disjoint areas are independent random variables. Probabilistic statements for this model can then be translated back to the original hyperbolic random graph model with a small penalty in certainty. A detailed explanation can be found in the full version of the paper [7].

**Probabilities.** Since we are analyzing a random graph model, our results are of probabilistic nature. To obtain meaningful statements, we show that they hold *with high probability* (with probability  $1 - \mathcal{O}(n^{-1})$ ), or *asymptotically almost surely* (with probability  $1 - o(1)$ ). The following Chernoff bound can be used to show that certain events occur with high probability.

► **Theorem 1** (Chernoff Bound [12, Theorem 1.1]). *Let  $X_1, \dots, X_n$  be independent random variables with  $X_i \in \{0, 1\}$  and let  $X$  be their sum. Then, for  $\varepsilon \in (0, 1)$*

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

Usually, it suffices to show that a random variable does not exceed an upper bound. The following corollary shows that a bound on the expected value suffices to obtain concentration.

► **Corollary 2.** *Let  $X_1, \dots, X_n$  be independent random variables with  $X_i \in \{0, 1\}$ , let  $X$  be their sum, and let  $f(n)$  be an upper bound on  $\mathbb{E}[X]$ . Then, for  $\varepsilon \in (0, 1)$*

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

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

► **Lemma 3** ([22, Lemma 2.3]). *Let  $x \in \mathbb{R}$  with  $x = \pm o(1)$ . Then,  $1/(1 + x) = 1 - \Theta(x)$ .*

### 3 An Improved Greedy Algorithm

Previous insights about solving the vertex cover problem on hyperbolic random graphs are based on the fact that the *dominance reduction rule* reduces the graph to a remainder of simple structure [5]. 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 [5, Lemma 5]. 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 4.** *Let  $G$  be a hyperbolic random graph on  $n$  vertices. Given the radii of the vertices, an approximate vertex cover of  $G$  can be computed in time  $\mathcal{O}(m \log(n))$ , such that the approximation ratio is  $(1 + o(1))$  asymptotically almost surely.*

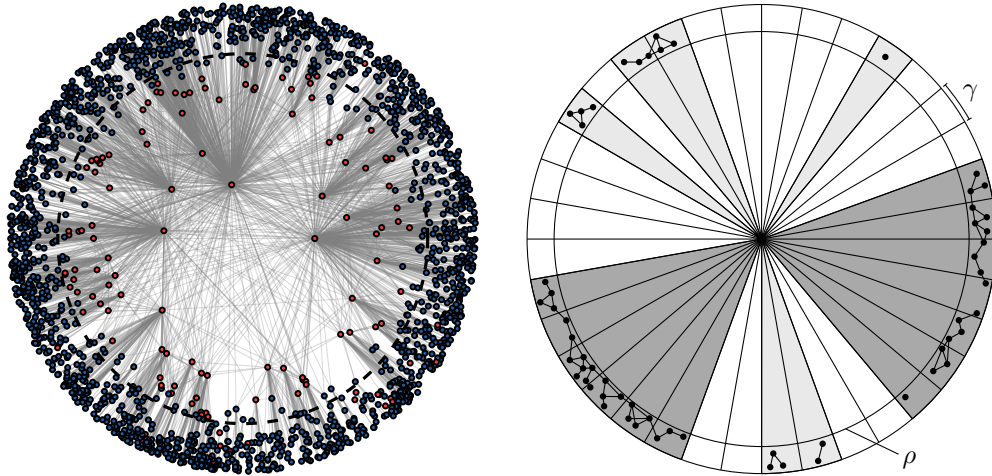
Consider the following greedy algorithm that computes an approximation of a minimum vertex cover on hyperbolic random graphs. We iterate the vertices in order of increasing radius. Each encountered vertex  $v$  is added to the cover and removed from the graph. After each step, we then identify the connected 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 tradeoff 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 [17]. 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.

### 4 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:  $V_{\text{Greedy}}$  and  $V_{\text{Exact}}$ , denoting the vertices that were added greedily and the ones contained in small separated components that were solved exactly, respectively (see Figure 1 (left)). Clearly, we



■ **Figure 1 (Left)** A hyperbolic random graph with 1942 vertices, average degree 7.7, and power-law exponent 2.6. The vertex sets  $V_{\text{Greedy}}$  and  $V_{\text{Exact}}$  are shown in red and blue, respectively. The dashed line shows a possible threshold radius  $\rho$ . **(Right)** The disk is divided into sectors of equal width  $\gamma$ . Consecutive non-empty sectors form a run (grey). Narrow runs (light grey) consist of few sectors. Wide runs (dark grey) consist of many sectors. Each dark grey sector is a widening sector.

obtain a valid vertex cover for the whole graph, if we take all vertices in  $V_{\text{Greedy}}$  together with a vertex cover  $C_{\text{Exact}}$  of  $G[V_{\text{Exact}}]$ . Then, the approximation ratio is given by the quotient  $\delta = (|V_{\text{Greedy}}| + |C_{\text{Exact}}|)/|C_{\text{OPT}}|$ , where  $C_{\text{OPT}}$  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  $|V_{\text{Greedy}}| \in o(|C_{\text{OPT}}|)$  in order to obtain the claimed approximation factor of  $1 + o(1)$ .

To bound the size of  $V_{\text{Greedy}}$ , 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  $V_{\text{Exact}}$ ), 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 a threshold radius  $\rho$  in the hyperbolic disk (see Figure 1). Therefore, we divide the hyperbolic disk into two regions: an *inner disk* and an *outer band*, containing vertices with radii below and above  $\rho$ , respectively. The threshold  $\rho$  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. 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 [10], this implies  $|V_{\text{Greedy}}| \in o(|C_{\text{OPT}}|)$ .

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 1 (right)). The foundation of this analysis is the delicate

interplay between the angular width  $\gamma$  of these sectors and the threshold  $\rho$  that defines the outer band. Recall that  $\rho$  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) were added greedily. Therefore, if we choose  $\rho$  too large, we overestimate the actual number of greedily added vertices by too much. As a consequence, we want to choose  $\rho$  as small as possible. However, this conflicts our intentions for the choice of  $\gamma$  and its impact on  $\rho$ . Recall that the maximum angular distance between two vertices such that they are adjacent increases with decreasing radii (Equation (2)). Thus, in order to avoid edges that extend beyond an angular width of  $\gamma$ , we need to ensure that the radii of the vertices in the outer band are sufficiently large. That is, decreasing  $\gamma$  requires increasing  $\rho$ . However, we want to make  $\gamma$  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,  $\gamma$  and  $\rho$  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  $\tau$  and  $n$ . To this end, we adjust the angular width of the sectors using a function  $\gamma(n, \tau)$ , which is defined as

$$\gamma(n, \tau) = \log(\tau \log^{(2)}(n) / (2 \log^{(3)}(n)^2)),$$

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.

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. This is mainly done by considering the random variables that denote the numbers of vertices in certain areas of the disk. We give proofs for their expected values throughout the paper. Tight concentration bounds can then be obtained using the previously mentioned Chernoff bound or, when the considered random variables are more involved, the method of (typical) bounded differences. These proofs can be found in the full version of the paper [7].

## 4.1 The Inner Disk

The inner disk  $\mathcal{I}$  contains all vertices whose radius is below the threshold  $\rho$ . The number of them that are added to the cover greedily is bounded by the number of all vertices in  $\mathcal{I}$ .

► **Lemma 5.** *Let  $G$  be a hyperbolic random graph on  $n$  vertices with power-law exponent  $\beta = 2\alpha + 1$ . With high probability, the number of vertices in  $\mathcal{I}$  is in  $\mathcal{O}(n \cdot \gamma(n, \tau)^{-\alpha})$ .*

**Proof.** We start by computing the expected number of vertices in  $\mathcal{I}$  and show concentration afterwards. To this end, we first compute the measure  $\mu(\mathcal{I})$ . The measure of a disk of radius  $r$  that is centered at the origin is given by  $e^{-\alpha(R-r)}(1 + o(1))$  [17, Lemma 3.2]. Consequently, the expected number of vertices in  $\mathcal{I}$  is

$$\mathbb{E}[|V(\mathcal{I})|] = n\mu(\mathcal{I}) = \mathcal{O}(ne^{-\alpha(R-\rho)}) = \mathcal{O}(ne^{-\alpha \log(\pi/2 \cdot e^{C/2} \gamma(n, \tau))}) = \mathcal{O}(n \cdot \gamma(n, \tau)^{-\alpha}).$$

Since  $\gamma(n, \tau) = \mathcal{O}(\log^{(3)}(n))$ , this bound on  $\mathbb{E}[|V(\mathcal{I})|]$  is  $\omega(\log(n))$ , and we can apply the Chernoff bound in Corollary 2 to conclude that  $|V(\mathcal{I})| = \mathcal{O}(n \cdot \gamma(n, \tau)^{-\alpha})$  holds with probability  $1 - \mathcal{O}(n^{-c})$  for any  $c > 0$ . ◀

Since  $\gamma(n, \tau) = \omega(1)$ , Lemma 5 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.

## 4.2 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  $\gamma = \theta(\rho, \rho) = \pi \cdot \gamma(n, \tau)/n \cdot (1 + o(1))$ , where  $\theta(\rho, \rho)$  denotes the maximum angular distance between two vertices with radii  $\rho$  to be adjacent (see Equation (2)). This division is depicted in Figure 1 (right). The choice of  $\gamma$  (combined with the choice of  $\rho$ ) 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 (grey in Figure 1 (right)). 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 that contain more than  $\tau \log \log(n)$  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.

### 4.2.1 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  $\mathcal{S}_k$  is empty 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 6.** *Let  $G$  be a hyperbolic random graph on  $n$  vertices. Then, the number of sectors of width  $\gamma = \theta(\rho, \rho)$  is  $n' = 2n/\gamma(n, \tau) \cdot (1 \pm o(1))$ .*

**Proof.** Since all sectors have equal angular width  $\gamma = \theta(\rho, \rho)$ , we can use Equation (2) to compute the total number of sectors as  $n' = 2\pi/\theta(\rho, \rho) = \pi e^{-R/2+\rho}(1 \pm \Theta(e^{R-2\rho}))^{-1}$ . By substituting  $\rho = R - \log(\pi/2 \cdot e^{C/2}\gamma(n, \tau))$  and  $R = 2 \log(n) + C$ , we obtain

$$n' = \frac{\pi e^{R/2}}{\pi/2 \cdot e^{C/2}\gamma(n, \tau)} (1 \pm \Theta(e^{-R}\gamma(n, \tau)^2))^{-1} = 2n/\gamma(n, \tau) \cdot (1 \pm \Theta((\gamma(n, \tau)/n)^2))^{-1}.$$

It remains to simplify the error term. Note that  $\gamma(n, \tau) = \mathcal{O}(\log^{(3)}(n))$ . Consequently, the error term is equivalent to  $(1 \pm o(1))^{-1}$ . Finally, it holds that  $1/(1+x) = 1 - \Theta(x)$  is valid for  $x = \pm o(1)$ , according to Lemma 3. ◀

► **Lemma 7.** *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*

$$1 - e^{-\gamma(n, \tau)/4} \leq \Pr[V(\mathcal{S}) \neq \emptyset] \leq e^{-\left(e^{-\gamma(n, \tau)}\right)}.$$

We are now ready to bound 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 8** ([23, Proposition 4.3<sup>1</sup>]). *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  $k \in \{1, \dots, n'\}$ . 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 2) and subsequently derive a probabilistic bound on  $W$  for hyperbolic random graphs.

► **Lemma 9.** *Let  $G$  be a hyperbolic random graph with  $n$  vertices in expectation and let  $W$  denote the number of widening sectors. Then,*

$$\mathbb{E}[W] \leq \frac{2^{1/4} \cdot \tau^{3/4} \cdot n}{\gamma(n, \tau) \cdot \log^{(2)}(n)^{1/4} \cdot \log^{(3)}(n)^{1/2}} (1 \pm o(1)).$$

**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 8. To this end, we use Lemma 6 to bound the total number of sectors  $n'$  and 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

<sup>1</sup> 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.

## 20:10 Approximating Vertex Cover on Networks with Underlying Hyperbolic Geometry

well as the complementary probability  $q = 1 - p \leq e^{-\gamma(n,\tau)/4}$  using Lemma 7. We obtain

$$\begin{aligned} \mathbb{E}[W] &= n' p^{(w+1)} ((w+1)q + p) \\ &\leq \frac{2n}{\gamma(n,\tau)} (1 \pm o(1)) \cdot e^{-((w+1)e^{-\gamma(n,\tau)})} \cdot \left( (w+1)e^{-\frac{\gamma(n,\tau)}{4}} + 1 \right) \\ &\leq \frac{2n}{\gamma(n,\tau)} e^{(-e^{\gamma(n,\tau)} \log^{(3)}(n) e^{-\gamma(n,\tau)})} \left( (e^{\gamma(n,\tau)} \log^{(3)}(n) + 1) e^{-\frac{\gamma(n,\tau)}{4}} + 1 \right) (1 \pm o(1)) \\ &= \frac{2ne^{3/4 \cdot \gamma(n,\tau)} \log^{(3)}(n)}{\gamma(n,\tau) \cdot \log^{(2)}(n)} \left( 1 + \frac{1}{e^{\gamma(n,\tau)} \log^{(3)}(n)} + \frac{1}{e^{3/4 \cdot \gamma(n,\tau)} \log^{(3)}(n)} \right) (1 \pm o(1)). \end{aligned}$$

Since  $\gamma(n,\tau) = \omega(1)$ , the first error term can be simplified as  $(1 + o(1))$ . Additionally, we can substitute  $\gamma(n,\tau) = \log(\tau \log^{(2)}(n)) / (2 \log^{(3)}(n^2))$  to obtain

$$\mathbb{E}[W] \leq 2^{1/4} \frac{\tau^{3/4} \cdot n \cdot \log^{(3)}(n)}{\gamma(n,\tau) \cdot \log^{(2)}(n)} \cdot \frac{\log^{(2)}(n)^{3/4}}{\log^{(3)}(n)^{3/2}} \cdot (1 \pm o(1)).$$

Further simplification then yields the claim.  $\blacktriangleleft$

**Concentration Bound on the Number of Widening Sectors.** Lemma 9 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 random variables can be bounded with high probability in general. Afterwards, we show that these conditions are met for our application.

► **Lemma 10.** *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  $g(n') = \omega(w \sqrt{n' \log(n')})$  is an upper bound on  $\mathbb{E}[W]$ , then  $W = \mathcal{O}(g(n'))$  holds with probability  $1 - \mathcal{O}((n')^{-c})$  for any constant  $c$ .*

► **Lemma 11.** *Let  $G$  be a hyperbolic random graph on  $n$  vertices. Then, with probability  $1 - \mathcal{O}(n^{-c})$  for any constant  $c > 0$ , the number of widening sectors  $W$  is bounded by*

$$W = \mathcal{O} \left( \frac{\tau^{3/4} \cdot n}{\gamma(n,\tau) \cdot \log^{(2)}(n)^{1/4} \cdot \log^{(3)}(n)^{1/2}} \right).$$

**Number of Vertices in Wide Runs.** Let  $\mathcal{W}$  denote the area of the disk covered by all widening sectors. By Lemma 11 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 12.** *Let  $G$  be a hyperbolic random graph on  $n$  vertices. Then, with high probability, the number of vertices in wide runs is bounded by*

$$|V(\mathcal{W})| = \mathcal{O} \left( \frac{\tau^{3/4} \cdot n}{\log^{(2)}(n)^{1/4} \cdot \log^{(3)}(n)^{1/2}} \right).$$

It remains to bound the number of vertices in large components contained in narrow runs.

### 4.2.2 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.

► **Lemma 13.** *Let  $G$  be a hyperbolic random graph on  $n$  vertices and let  $\mathcal{R}$  be a narrow run. Then,  $\mathbb{E}[|V(\mathcal{R})|] \leq 1/2 \cdot e^{\gamma(n,\tau)} \log^{(3)}(n) \gamma(n,\tau) (1 \pm o(1))$ .*

**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  $\mathbb{E}[|V(\mathcal{R})|] \leq nw/n'$ . As  $n' = 2n/\gamma(n,\tau)(1 \pm o(1))$  according to Lemma 6, we have

$$\mathbb{E}[|V(\mathcal{R})|] \leq 1/2 \cdot e^{\gamma(n,\tau)} \log^{(3)}(n) \gamma(n,\tau) (1 \pm o(1))^{-1}.$$

Since  $1/(1+x) = 1 - \Theta(x)$  for  $x = \pm o(1)$  (Lemma 3), 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 14.** *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  $\Pr[|V(\mathcal{R})| = k] \leq e^{-k/18}$ .*

We are now ready to compute the expected number of vertices in all large narrow runs.

► **Lemma 15.** *Let  $G$  be a hyperbolic random graph. Then, the expected number of vertices in all large narrow runs is bounded by*

$$\mathbb{E}[N] = \mathcal{O} \left( \frac{\tau \cdot n \cdot \log^{(2)}(n)}{\gamma(n,\tau) \log(n)^{\tau/18}} \right).$$

**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,

$$N = \sum_{i=1}^{n''} |V(\mathcal{R}_i)| \cdot \mathbb{1}_{|V(\mathcal{R}_i)| > \tau \log^{(2)}(n)}.$$

Consequently, the expected value of  $N$  is given by

$$\mathbb{E}[N] = \sum_{i=1}^{n''} \mathbb{E} \left[ |V(\mathcal{R}_i)| \cdot \mathbb{1}_{|V(\mathcal{R}_i)| > \tau \log^{(2)}(n)} \right] = \sum_{i=1}^{n''} \sum_{k=\tau \log^{(2)}(n)+1}^n k \cdot \Pr[|V(\mathcal{R}_i)| = k].$$

Lemma 14 gives a valid upper bound on  $\Pr[|V(\mathcal{R}_i)| = k]$  for all  $i \in \{1, \dots, n''\}$ . Furthermore, the number of narrow runs  $n''$  is bounded by the 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

$$\begin{aligned} \mathbb{E}[N] &\leq n' \int_{\tau \log^{(2)}(n)}^n k e^{-\frac{k}{18}} dk \\ &\leq n' \left[ 18e^{-\tau/18 \log^{(2)}(n)} (\tau \log^{(2)}(n) + 18) - 18e^{-n/18} (n + 18) \right] \\ &\leq 18n' \cdot \frac{\tau \log^{(2)}(n) + 18}{\log(n)^{\tau/18}}, \end{aligned}$$

where the last inequality holds since  $e^{-n/18}(n+18) \geq 0$ . Substituting  $n' = 2n/\gamma(n, \tau)(1 \pm o(1))$  (Lemma 6) and more simplification yield the claim.  $\blacktriangleleft$

**Concentration Bound on the Number of Vertices in Large Narrow Runs.** To show that the actual number of vertices in large narrow runs is not much larger than the expected value, we apply the method of typical bounded differences [29]. (See the full version of the paper for a detailed explanation [7].) To this end, we consider  $N$  as a function of the positions of the vertices and bound the effect that changing a single position can have on this function. It is easy to see, that this effect is the largest, when the change splits a wide run  $\mathcal{R}$  into two large narrow runs. If  $\mathcal{R}$  contained a lot of vertices, the impact on  $N$  is large. However, since the vertices are distributed uniformly, it is very unlikely that a run that can be split into two narrow runs contains many vertices.

► **Lemma 16.** *Let  $G$  be a hyperbolic random graph. Then, each run of length at most  $2w + 1$  contains at most  $\mathcal{O}(\log(n))$  vertices with probability  $1 - \mathcal{O}(n^{-c})$  for any constant  $c$ .*

The method of typical bounded differences now allows us to focus on this case and to milder 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 17.** *Let  $G$  be a hyperbolic random graph. Then, with high probability, the number of vertices in large narrow runs is bounded by*

$$N = \mathcal{O} \left( \frac{\tau \cdot n \cdot \log^{(2)}(n)}{\gamma(n, \tau) \log(n)^{\tau/18}} \right).$$

### 4.3 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  $\tau$  can be used to obtain a tradeoff between approximation performance and running time.

► **Theorem 18.** *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  $\mathcal{O}(m \log(n)^\tau + n \log(n))$ , such that the approximation factor is  $(1 + \mathcal{O}(\gamma(n, \tau)^{-\alpha}))$  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  $\mathcal{O}(n \log(n))$ . 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  $\mathcal{O}(1)$  and  $\mathcal{O}(\deg(v))$ , 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  $\mathcal{O}(\log \log(n)^2)$ . 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  $1.1996^{n_c} \cdot n_c^{\mathcal{O}(1)}$  [30]. Since  $n_c^{\mathcal{O}(1)} = \mathcal{O}((e/1.1996)^{n_c})$ , this running time is bounded by  $\mathcal{O}(e^{n_c}) = \mathcal{O}(\log(n)^\tau)$ . Consequently, the time required to process each neighbor of  $v$  is  $\mathcal{O}(\log(n)^\tau)$ . 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 summing the running times of all three steps over all vertices, which yields

$$\begin{aligned} T(n, m, \tau) &= \sum_{v \in V} \mathcal{O}(1) + \mathcal{O}(\deg(v)) + \deg(v) \cdot \mathcal{O}(\log(n)^\tau) \\ &= \mathcal{O} \left( \log(n)^\tau \cdot \sum_{v \in V} \deg(v) \right) = \mathcal{O}(m \log(n)^\tau). \end{aligned}$$

*Approximation Ratio.* As argued before, we obtain a valid vertex cover for the whole graph, if we take all vertices in  $V_{\text{Greedy}}$  together with a vertex cover  $C_{\text{Exact}}$  of  $G[V_{\text{Exact}}]$ . The approximation ratio of the resulting cover is then given by the quotient

$$\delta = \frac{|V_{\text{Greedy}}| + |C_{\text{Exact}}|}{|C_{\text{OPT}}|},$$

where  $C_{\text{OPT}}$  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  $\delta \leq 1 + |V_{\text{Greedy}}|/|C_{\text{OPT}}|$ . To bound the number of vertices in  $V_{\text{Greedy}}$ , we add the number of vertices in the inner disk  $\mathcal{I}$ , 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,

$$\delta \leq 1 + \frac{|V(\mathcal{I})| + |V(\mathcal{W})| + |V(\mathcal{N})|}{|C_{\text{OPT}}|}.$$

Upper bounds on  $|V(\mathcal{I})|$ ,  $|V(\mathcal{W})|$ , and  $|V(\mathcal{N})|$  that hold with high probability are given by Lemmas 5, 12, and 17, respectively. Furthermore, it was previously shown that the size of a minimum vertex cover on a hyperbolic random graph is  $|C_{\text{OPT}}| = \Omega(n)$ , asymptotically almost surely [10, Theorems 4.10 and 5.8]. We obtain

$$\delta = 1 + \mathcal{O} \left( \frac{1}{\gamma(n, \tau)^\alpha} + \frac{\tau^{3/4}}{\log^{(2)}(n)^{1/4} \cdot \log^{(3)}(n)^{1/2}} + \frac{\tau \cdot \log^{(2)}(n)}{\gamma(n, \tau) \log(n)^{\tau/18}} \right).$$

Since  $\gamma(n, \tau) = \mathcal{O}(\log^{(3)}(n))$ , the first summand dominates asymptotically.  $\blacktriangleleft$

**► Theorem 4.** *Let  $G$  be a hyperbolic random graph on  $n$  vertices. Given the radii of the vertices, an approximate vertex cover of  $G$  can be computed in time  $\mathcal{O}(m \log(n))$ , such that the approximation ratio is  $(1 + o(1))$  asymptotically almost surely.*

**Proof.** By Theorem 18 we can compute an approximate vertex cover in time  $\mathcal{O}(n \log(n) + m \log(n)^\tau)$ , such that the approximation factor is  $1 + \mathcal{O}(\gamma(n, \tau)^{-\alpha})$ , asymptotically almost surely. We obtain the claimed bound on the running time by choosing  $\tau = 1$ . Furthermore, since  $\gamma(n, 1) = \omega(1)$  and  $\alpha \in (1/2, 1)$ , the resulting approximation factor is  $(1 + o(1))$ .  $\blacktriangleleft$

---

References

---

- 1 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 *Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics*, pages 62–69, 2004.
- 2 Takuya Akiba and Yoichi Iwata. Branch-and-reduce exponential/FPT algorithms in practice: A case study of vertex cover. *Theor. Comput. Sci.*, 609:211–225, 2016. doi:10.1016/j.tcs.2015.09.023.
- 3 Eric Angel, Romain Campigotto, and Christian Laforest. Implementation and comparison of heuristics for the vertex cover problem on huge graphs. In *Experimental Algorithms*, pages 39–50, 2012.
- 4 Yonatan Bilu and Nathan Linial. Are stable instances easy? *Comb. Probab. Comput.*, 21(5):643–660, 2012. doi:10.1017/S0963548312000193.
- 5 Thomas Bläsius, Philipp Fischbeck, Tobias Friedrich, and Maximilian Katzmann. Solving Vertex Cover in Polynomial Time on Hyperbolic Random Graphs. In *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*, pages 25:1–25:14, 2020. doi:10.4230/LIPIcs.STACS.2020.25.
- 6 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)*, pages 20:1–20:14, 2018. doi:10.4230/LIPIcs.ICALP.2018.20.
- 7 Thomas Bläsius, Tobias Friedrich, and Maximilian Katzmann. Efficiently approximating vertex cover on scale-free networks with underlying hyperbolic geometry. *CoRR*, abs/2010.02787, 2020. URL: <https://arxiv.org/abs/2010.02787>.
- 8 Marián Boguná, Fragkiskos Papadopoulos, and Dmitri Krioukov. Sustaining the internet with hyperbolic mapping. *Nature Communications*, 1:62, 2010. doi:10.1038/ncomms1063.
- 9 Vaggos Chatziafratis, Tim Roughgarden, and Jan Vondrak. Stability and Recovery for Independence Systems. In *25th Annual European Symposium on Algorithms (ESA 2017)*, pages 26:1–26:15, 2017. doi:10.4230/LIPIcs.ESA.2017.26.
- 10 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 (FSTTCS 2016)*, pages 33:1–33:15, 2016. doi:10.4230/LIPIcs.FSTTCS.2016.33.
- 11 Mariana O. Da Silva, Gustavo A. Gimenez-Lugo, and Murilo V. G. Da Silva. Vertex cover in complex networks. *Int. J. Mod. Phys. C*, 24(11):1350078, 2013. doi:10.1142/S0129183113500782.
- 12 Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2012.
- 13 Leah Epstein, Asaf Levin, and Gerhard J. Woeginger. Vertex cover meets scheduling. *Algorithmica*, 74:1148–1173, 2016. doi:10.1007/s00453-015-9992-y.
- 14 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*, 1:2931–2937, 2007.
- 15 Guillermo García-Pérez, Marián Boguná, Antoine Allard, and M. Serrano. The hidden hyperbolic geometry of international trade: World trade atlas 1870–2013. *Scientific Reports*, 6:33441, 2016. doi:10.1038/srep33441.
- 16 Mikael Gast and Mathias Hauptmann. Approximability of the vertex cover problem in power-law graphs. *Theoretical Computer Science*, 516:60–70, 2014. doi:10.1016/j.tcs.2013.11.004.

- 17 Luca Gugelmann, Konstantinos Panagiotou, and Ueli Peter. Random hyperbolic graphs: Degree sequence and clustering. In *Automata, Languages, and Programming*, pages 573–585. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-31585-5\_51.
- 18 George Karakostas. A better approximation ratio for the vertex cover problem. *ACM Trans. Algorithms*, 5(4):41:1–41:8, 2009. doi:10.1145/1597036.1597045.
- 19 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.
- 20 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within  $2 - \epsilon$ . *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. doi:10.1016/j.jcss.2007.06.019.
- 21 Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, 2010. doi:10.1103/PhysRevE.82.036106.
- 22 Anton Krohmer. *Structures & algorithms in Hyperbolic Random Graphs*. doctoralthesis, University of Potsdam, 2016.
- 23 Frosso S. Makri, Andreas N. Philippou, and Zaharias M. Psillakis. Success run statistics defined on an urn model. *Advances in Applied Probability*, 39(4):991–1019, 2007. doi:10.1239/aap/1198177236.
- 24 Tobias Müller and Merlijn Staps. The diameter of KPKVB random graphs. *Advances in Applied Probability*, 51:358–377, 2019. doi:10.1017/apr.2019.23.
- 25 Kihong Park and Walter Williger. *The Internet As a Large-Scale Complex System*. Oxford University Press, Inc., 2005.
- 26 K. Subhash, D. Minzer, and M. Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 592–601, 2018.
- 27 Kevin Verbeek and Subhash Suri. Metric embedding, hyperbolic space, and social networks. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG’14, page 501–510, 2014. doi:10.1145/2582112.2582139.
- 28 André L. Vignatti and Murilo V.G. da Silva. Minimum vertex cover in generalized random graphs with power law degree distribution. *Theoretical Computer Science*, 647:101–111, 2016. doi:10.1016/j.tcs.2016.08.002.
- 29 Lutz Warnke. On the method of typical bounded differences. *Combinatorics, Probability and Computing*, 25(2):269–299, 2016. doi:10.1017/S0963548315000103.
- 30 Mingyu Xiao and Hiroshi Nagamochi. Exact algorithms for maximum independent set. *Inf. Comput.*, 255:126–146, 2017. doi:10.1016/j.ic.2017.06.001.