Solving Vertex Cover in Polynomial Time on Hyperbolic Random Graphs

Thomas Bläsius
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany

Philipp Fischbeck
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany

Tobias Friedrich
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany

Maximilian Katzmann
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany

Abstract

The VertexCover problem is proven to be computationally hard in different ways: It is NP-complete to find an optimal solution and even NP-hard to find an approximation with reasonable factors. In contrast, recent experiments suggest that on many real-world networks the run time to solve VertexCover is way smaller than even the best known FPT-approaches can explain. Similarly, greedy algorithms deliver very good approximations to the optimal solution in practice.

We link these observations to two properties that are observed in many real-world networks, namely a heterogeneous degree distribution and high clustering. To formalize these properties and explain the observed behavior, we analyze how a branch-and-reduce algorithm performs on hyperbolic random graphs, which have become increasingly popular for modeling real-world networks. In fact, we are able to show that the VertexCover 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. Since these predictions of the model are interesting in their own right, we conducted experiments on real-world networks showing that these properties are also observed in practice. When utilizing the same structural properties in an adaptive greedy algorithm, further experiments suggest that this leads to even better approximations than the standard greedy approach on real instances.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Random network models; Mathematics of computing → Random graphs

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

Digital Object Identifier 10.4230/LIPIcs.STACS.2020.23

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

VertexCover is a fundamental NP-complete graph problem. For a given undirected graph G on n vertices the goal is to find the smallest vertex subset S, such that each edge in G is incident to at least one vertex in S. Since, by definition, there can be no edge between two vertices outside of S, these remaining vertices form an independent set. Therefore, one can easily derive a maximal independent set from a minimal vertex cover and vice versa.

Due to its NP-completeness there is probably no polynomial time algorithm for solving VertexCover. The best known algorithm for IndependentSet runs in $1.996^n \operatorname{poly}(n)$ [22]. To analyze the complexity of VertexCover on a finer scale, several parameterized solutions have been proposed. One can determine whether a graph G 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 branch-and-reduce technique yields a simple $\mathcal{O}(2^k \operatorname{poly}(n))$ algorithm, where the exponential portion comes from the branching. The best known FPT algorithm runs in $\mathcal{O}(1.2738^k + kn)$ time [7], and unless ETH fails, there can be no $2^{o(\sqrt{k})} \operatorname{poly}(n)$ algorithm [8].

While these FPT approaches promise relatively small running times if the considered network has a small vertex cover, the cover is large for many real-world networks. Nevertheless, it was recently observed that applying a branch-and-reduce technique on real instances is very efficient [2]. 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. Most notably, applying the *dominance reduction rule*, which eliminates vertices whose neighborhood contains a vertex together with its neighborhood, 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* (the network contains many vertices of small degree and few vertices of high degree) and *high clustering* (the neighbors of a vertex are likely to be neighbors themselves).

We formalize these key properties using *hyperbolic random graphs* to analyze the performance of the dominance rule. Introduced by Krioukov et al. [17], hyperbolic random graphs are obtained by randomly distributing nodes in the hyperbolic plane and connecting any two that are geometrically close. The resulting graphs feature a power-law degree distribution and high clustering [14, 17] (the two desired properties) which can be tuned using parameters of the model. Additionally, the generated networks have a small diameter [13]. All of these properties have been observed in many real-world networks such as the internet, social networks, as well as biological networks like protein-protein interaction networks. Furthermore, Boguná, Papadopoulos, and Krioukov showed that the internet can be embedded into the hyperbolic plane such that routing packages between network participants greedily works very well [5], indicating that this network naturally fits into the hyperbolic space.

By making use of the underlying geometry, we show that VertexCover can be solved in polynomial time on hyperbolic random graphs, with high probability. This is done by showing that the dominance reduction rule reduces a hyperbolic random graph to a remainder with small pathwidth on which VertexCover can then be solved efficiently. We note that, while our analysis makes use of the underlying hyperbolic geometry, the algorithm itself is oblivious to it. Our analysis provides an explanation for why VertexCover can be solved efficiently on practical instances. Besides the running time itself the model predicts certain structural properties that also point us to an adapted greedy algorithm that achieves better approximation ratios while still being very efficient. We conducted experiments indicating that these predictions (concerning the structural properties and improved approximation) actually match the real world for a significant fraction of networks.

2 Preliminaries

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

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

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

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 2π.

We use $B_p(r)$ to denote a disk of radius r centered at p, i.e., the set of points with hyperbolic distance at most r to p. Such a disk has an area of $2\pi(\cosh(r) - 1)$ and circumference $2\pi \sinh(r)$. Thus, the area and the circumference of a disk in the hyperbolic plane grow exponentially with its radius. In contrast, this growth is polynomial in Euclidean space. Therefore, representing hyperbolic shapes in the Euclidean geometry results in a distortion. In the *native representation*, used in our figures, circles can appear teardrop-shaped (see Figure 1).

Hyperbolic Random Graphs. Hyperbolic random graphs are obtained by distributing n points uniformly at random within the disk $B_O(R)$ and connecting any two of them if and only if their hyperbolic distance is at most R. The disk radius R (which matches the connection threshold) is defined as $R = 2 \log(8n/(\pi\bar{\kappa}))$, where $\bar{\kappa}$ is a constant describing the desired average degree of the generated network. The coordinates for 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 to the probability density function

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

for $r \in [0, R]$. For $r > R$, $f(r) = 0$. The constant $\alpha \in (1/2, 1)$ is used to tune the power-law exponent $\beta = 2\alpha + 1$ of the degree distribution of the generated network. Note that we obtain power-law exponents $\beta \in (2, 3)$. Exponents outside of this range are atypical for hyperbolic random graphs. On the one hand, for $\beta < 2$ the average degree of the generated networks are divergent. On the other hand, for $\beta > 3$ hyperbolic random graphs degenerate:

They decompose into smaller components, none having a size linear in n. The obtained graphs have logarithmic tree width [4], meaning the VertexCover problem can be solved efficiently in that case.

The probability for a given vertex to lie in a certain area A of the disk is given by its probability measure $\mu(A) = \int_A f(r)\mathrm{d}r$. 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 [14, Lemma 6]

$$\theta(r(u), r(v)) = \arccos\left(\frac{\cosh(r(u))\cosh(r(v)) - \cosh(R)}{\sinh(r(u))\sinh(r(v))}\right)$$
$$= 2e^{(R-r(u)-r(v))/2}(1 + \Theta(e^{R-r(u)-r(v)})). \quad (2)$$

Interval Graphs and Circular Arc 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. Circular arc graphs are a superclass of interval graphs, where 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 $2k$.

Treewidth and Pathwidth. A *tree decomposition* of a graph G is a tree T where each tree node represents a subset of the vertices of G called *bag*, and the following 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. The *path decomposition* of a graph is defined analogously to the tree decomposition, with the constraint that the tree has to be a path. Additionally, as for the treewidth, the *pathwidth* of a graph G, denoted by $\mathrm{pw}(G)$, is the minimum width over all path decompositions of G. Clearly the pathwidth is an upper bound on the treewidth. 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 [8, 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.

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* (for short *whp.*), i.e., with probability $1 - \mathcal{O}(n^{-1})$. The following Chernoff bound is a useful tool for showing that certain events occur with high probability.

▶ **Theorem 1** (Chernoff Bound [11, A.1])**.** *Let $X_1, \ldots, X_n$ be independent random variables with $X_i \in \{0, 1\}$ and let X be their sum. Let $f(n) = \Omega(\log(n))$. If $f(n)$ is an upper bound for $\mathbb{E}[X]$, then for each constant c there exists a constant c' such that $X \leq c' f(n)$ holds with probability $1 - \mathcal{O}(n^{-c})$.*

3 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 was previously defined for the IndependentSet problem [12], and later used for VertexCover in the experiments by Akiba and Iwata [2]. Formally, vertex u *dominates* a neighbor $v \in N(u)$ if $(N(v) \setminus \{u\}) \subseteq N(u)$, i.e., all neighbors of v are also neighbors of u. We say 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 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 G, it is thus safe to assume that u is part of the solution and to reduce the search to $G[V \setminus \{u\}]$.

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

▶ **Theorem 2.** *Let G be a hyperbolic random graph on n vertices. Then the* VertexCover *problem on G can be solved in* $\mathrm{poly}(n)$ *time, with high probability.*

The proof of Theorem 2 consists of two parts that make use of the underlying hyperbolic geometry. In the first part, we show that applying the dominance reduction rule, removes all vertices in the inner part of the hyperbolic disk, with high probability. 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. We prove that this subgraph has a small pathwidth, by showing that there is a circular arc supergraph with a small interval width. Consequently, a tree decomposition of this subgraph can be computed efficiently. Finally, we obtain a polynomial time algorithm for VertexCover by first applying the reduction rules and afterwards solving VertexCover on the remaining subgraph using the tree decomposition of small width.

3.1 Dominance on Hyperbolic Random Graphs

Recall that a hyperbolic random graph is obtained by distributing n vertices in a hyperbolic disk $B_O(R)$ and that any two are connected if their distance is at most R. Consequently, one can imagine the neighborhood of a vertex u as another disk $B_u(R)$. Vertex u dominates another vertex v if its neighborhood disk completely contains that of v (both constrained to $B_O(R)$), as depicted in Figure 1 left. We define the *dominance area* $D(u)$ of u to be the area containing all such vertices v. That is, $D(u) = \{p \in B_O(R) \mid B_p(R) \cap B_O(R) \subseteq B_u(R) \cap B_O(R)\}$. The result is illustrated in Figure 1 right. We note that it is sufficient for a vertex v to lie in $D(u)$ in order to be dominated by u, however, it is not necessary.

Given the radius $r(u)$ of vertex u we can now compute the probability that u dominates another vertex, i.e., the probability that at least one vertex lies in $D(u)$, by determining

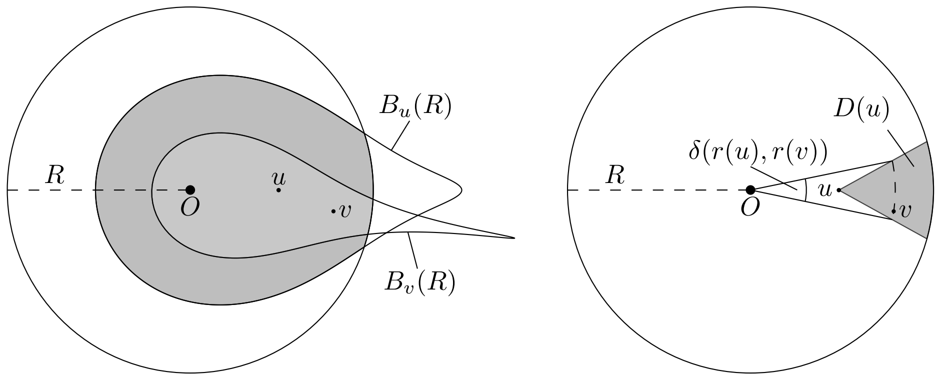

Figure 1 Left: Vertex u dominates vertex v, as $B_v(R) \cap B_O(R)$ (light gray) is completely contained in $B_u(R) \cap B_O(R)$ (gray). Right: All vertices that lie in $D(u)$ are dominated by u.

the measure $\mu(D(u))$. To this end, we first define $\delta(r(u), r(v))$ to be the maximum angular distance between two nodes u and v such that v lies in $D(u)$.

▶ **Lemma 3.** *Let u, v be vertices with $r(u) \leq r(v)$. Then, $v \in D(u)$ if $\Delta_\varphi(u, v)$ is at most*

$$\delta(r(u), r(v)) = 2(e^{-r(u)/2} - e^{-r(v)/2}) + \Theta(e^{-3/2r(u)}) - \Theta(e^{-3/2r(v)}).$$

Proof. Without loss of generality we assume that $\varphi(u) = 0$. For now assume that $\varphi(v) = \varphi(u)$. Since $r(v) \geq r(u)$ we know that the intersections of the boundaries of $B_v(R)$ with $B_O(R)$ lie between those of $B_u(R)$ with $B_O(R)$, as is depicted in Figure 2. Now let i_u denote one of these intersections for $B_u(R)$ and $B_O(R)$, and let i_v denote the intersection for $B_v(R)$ and $B_O(R)$ that is on the same side of the ray through O and u as i_u. It is easy to see that the maximum angular distance between u and v such that $B_v(R) \cap B_O(R)$ is contained within $B_u(R) \cap B_O(R)$ is given by the angular distance between i_u and i_v. Therefore, v lies in the domination area of u if $\Delta_\varphi(u, v) \leq \Delta_\varphi(i_u, i_v)$.

Recall that $\theta(r(p), r(q))$ denotes the maximum angular distance such that $\mathrm{dist}(p, q) \leq R$, as defined in Equation (2). Since i_u and i_v have radius R and hyperbolic distance R to u and v, respectively, we know that their angular coordinates are $\theta(r(u), R)$ and $\theta(r(v), R)$, respectively. Consequently, the angular distance between i_u and i_v is given by

$$\begin{aligned}\delta(r(u), r(v)) &= \theta(r(u), R) - \theta(r(v), R) \\ &= 2(e^{-r(u)/2} - e^{-r(v)/2}) + \Theta(e^{-3/2r(u)}) - \Theta(e^{-3/2r(v)}).\end{aligned}$$ ◀

Using Lemma 3 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 $2/\pi e^{-r(u)/2}$, which is a constant fraction of the measure of the neighborhood disk of u which grows as $2\alpha/((\alpha - 1/2)\pi)e^{-r(u)/2}$ [14, Lemma 3.2]. Consequently, the expected number of nodes that u dominates is a constant fraction of the expected number of its neighbors.

▶ **Lemma 4.** *Let u be a node with radius $r(u) \geq R/2$. The probability for a given node to lie in $D(u)$ is given by*

$$\mu(D(u)) = \frac{2}{\pi} e^{-r(u)/2}(1 - \Theta(e^{-\alpha(R - r(u))})) \pm \mathcal{O}(1/n).$$

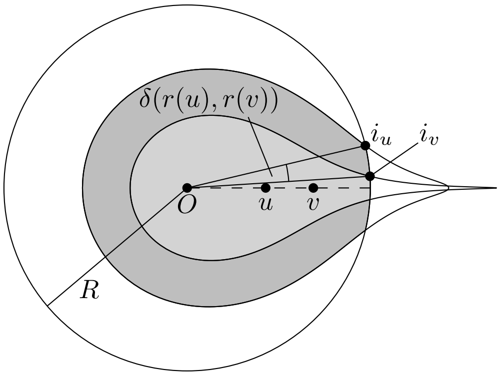

Figure 2 Vertex u dominates vertex v, with $r(u) \leq r(v)$, if $\Delta_\varphi(u, v) \leq \Delta_\varphi(i_u, i_v)$.

Proof. The probability for a given vertex v to lie in $D(u)$ is obtained by integrating the probability density (given by Equation (1)) over $D(u)$.

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

Since $r(u) \geq R/2$ and $r \in [r(u), R]$ we have $\Theta(e^{-3/2r(u)}) - \Theta(e^{-3/2r}) = \pm\mathcal{O}(e^{-3/4R})$ and $(1 + \Theta(e^{-\alpha R} - e^{-2\alpha r})) = (1 + \Theta(e^{-\alpha R}))$. Due to the linearity of integration, constant factors within the integrand can be moved out of the integral, which yields

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

The remaining integrals can be computed easily and we obtain

$$\begin{aligned}\mu(D(u)) &= \frac{2}{\pi}e^{-r(u)/2}(1+\Theta(e^{-\alpha R}))(1-e^{-\alpha(R-r(u))}) \\ &\quad - \frac{2\alpha}{(\alpha-1/2)\pi}e^{-R/2}(1+\Theta(e^{-\alpha R}))(1-e^{-(\alpha-1/2)(R-r(u))}) \\ &\quad \pm \mathcal{O}\left(e^{-3/4R}(1-e^{-\alpha(R-r(u))})\right).\end{aligned}$$

As $e^{-R/2} = \Theta(n^{-1})$ and $e^{-3/4R} = \Theta(n^{-3/2})$, simplifying the error terms 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 with high probability.

▶ **Lemma 5.** *Let G be a hyperbolic random graph with average degree $\bar{\kappa}$. Then there is a constant $c > 4/\bar{\kappa}$, 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 $D(u)$. To show this for any u with $r(u) \le \rho$, it suffices to show it for $r(u) = \rho$, since $D(u)$ increases with decreasing radius. To determine the probability that at least one vertex lies in $D(u)$, we use Lemma 4 and obtain

$$\mu(D(u)) = \frac{2}{\pi} e^{-\rho/2}(1 - \Theta(e^{-\alpha(R-\rho)})) \pm \mathcal{O}(1/n)$$
$$= \frac{2}{\pi} e^{-R/2+\log\log(n^c)}(1 - \Theta(e^{-2\alpha \log\log(n^c)})) \pm \mathcal{O}(1/n).$$

By substituting $R = 2\log(8n/(\pi\bar{\kappa}))$, we obtain $\mu(D(u)) = \bar{\kappa}/(4n)(c\log(n)(1 - o(1)) \pm \mathcal{O}(1))$. The probability of at least one node falling into the $D(u)$ is now given by

$$\Pr[\{v \in D(u)\} \neq \emptyset] = 1 - (1 - \mu(D(u)))^n \ge 1 - e^{-n\mu(D(u))} = 1 - \Theta(n^{-c\bar{\kappa}/4(1-o(1))}).$$

Consequently, for large enough n we can choose $c > 4/\bar{\kappa}$ such that the probability of a vertex at radius ρ being dominant is at least $1 - \Theta(n^{-2})$, allowing us to apply union bound. ◀

▶ **Corollary 6.** *Let G be a hyperbolic random graph and $c > 4/\bar{\kappa}$. With high probability, all vertices with radius at most $\rho = R - 2\log\log(n^c)$ are removed by the dominance rule.*

By Corollary 6 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 $B_O(R) \setminus B_O(\rho)$. It remains to show that the pathwidth of the subgraph induced by the vertices in the outer band is small.

3.2 Pathwidth in the Outer Band

In the following, we use $G_r = (V_r, E_r)$ to denote the induced subgraph of G that contains all vertices with radius at least r. To show that the pathwidth of G_ρ (the induced subgraph in the outer band) is small, we first show that there is a circular arc supergraph G_ρ^S of G_ρ with a small maximum clique. We use G^S 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 $I_v = [\varphi(v) - \theta(r(v), r(v)), \varphi(v) + \theta(r(v), r(v))]$. Intuitively, this means that the interval of a vertex contains a superset of all its neighbors that have a larger radius, as can be seen in Figure 3 left. The following lemma shows that G^S is actually a supergraph of G.

▶ **Lemma 7.** *Let $G = (V, E)$ be a hyperbolic random graph. Then G^S is a supergraph of G.*

Proof. Let $\{u, v\} \in E$ be any edge in G. To show that G^S is a supergraph of G we need to show that u and v are also adjacent in G^S, i.e., $I_u \cap I_v \neq \emptyset$. Without loss of generality assume $r(u) \le r(v)$. 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 $\theta(r(u), r(v)) \le \theta(r(u), r(u))$ for $r(u) \le r(v)$, we have $\Delta_\varphi(u, v) \le \theta(r(u), r(u))$. As I_u extends by $\theta(r(u), r(u))$ from $\varphi(u)$ in both directions, it follows that $\varphi(v) \in I_u$. ◀

It is easy to see that, after removing a vertex from G and G^S, G^S is still a supergraph of G. Consequently, G_ρ^S is a supergraph of G_ρ. It remains to show that G_ρ^S 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 the number of arcs that intersect a given angle which we set to 0 without loss of generality. Let A_r denote the area of the disk containing all vertices v with radius $r(v) \ge r$ whose interval I_v intersects 0, as illustrated in Figure 3 right. The following lemma describes the probability for a given vertex to lie in A_r.

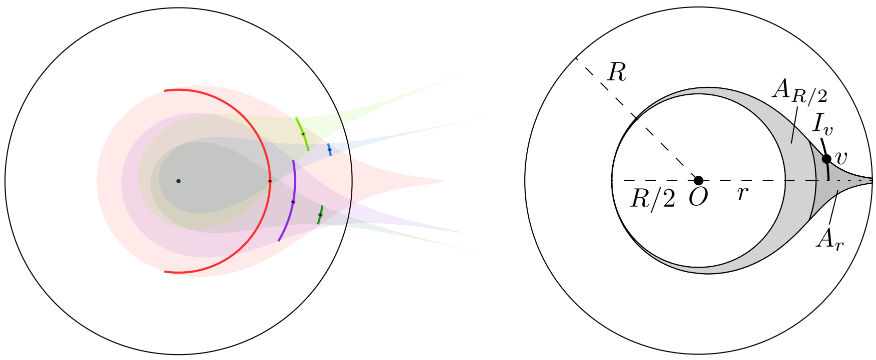

Figure 3 Left: The circular arcs representing the neighborhood of a vertex. For vertex v the area containing the whole neighborhood of v, as well as the circular arc I_v are drawn in the same color. Right: The area that contains the vertices whose arcs intersect angle 0. Area A_r contains all such vertices with radius at least r. Vertex v lies on the boundary of A_r and its interval I_v extends to 0.

▶ **Lemma 8.** *Let G be a hyperbolic random graph and let $r \geq R/2$. The probability for a given vertex to lie in A_r is bounded by*

$$\mu(A_r) \leq \frac{2\alpha}{(1-\alpha)\pi} e^{-(\alpha-1/2)R-(1-\alpha)r} \cdot \left(1 + \Theta(e^{-\alpha R} + e^{-(2r-R)} - e^{-(1-\alpha)(R-r)})\right).$$

Proof. We obtain the measure of A_r by integrating the probability density function over A_r. Following the definition of I_v for a vertex v, we can conclude that A_r includes all vertices v with radius $r(v) \geq r$ whose angular distance to 0 is at most $\theta(r(v), r(v))$. We obtain

$$\begin{aligned}\mu(A_r) &= \int_r^R 2 \int_0^{\theta(x,x)} f(x)\,\mathrm{d}\varphi\,\mathrm{d}x \\ &= 2\int_r^R 2e^{(R-2x)/2}(1 \pm \Theta(e^{R-2x})) \cdot \frac{\alpha}{2\pi} e^{-\alpha(R-x)}(1 + \Theta(e^{-\alpha R} - e^{-2\alpha x}))\,\mathrm{d}x.\end{aligned}$$

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

$$\mu(A_r) \leq \frac{2\alpha}{\pi} e^{-(\alpha-1/2)R}(1 + \Theta(e^{-\alpha R})) \int_r^R e^{-(1-\alpha)x}(1 + \Theta(e^{R-2x}))\,\mathrm{d}x.$$

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

$$\begin{aligned}\mu(A_r) &\leq \frac{2\alpha}{\pi} e^{-(\alpha-1/2)R}(1 + \Theta(e^{-\alpha R}))\left(\int_r^R e^{-(1-\alpha)x}\,\mathrm{d}x + \Theta\left(\int_r^R e^{-(1-\alpha)x+R-2x}\,\mathrm{d}x\right)\right) \\ &= \frac{2\alpha}{\pi} e^{-(\alpha-1/2)R}(1 + \Theta(e^{-\alpha R})) \\ &\quad \cdot \left(\frac{1}{1-\alpha} e^{-(1-\alpha)r}(1 - e^{-(1-\alpha)(R-r)}) + \Theta\left(e^R e^{-(3-\alpha)r}(1 - e^{-(3-\alpha)(R-r)})\right)\right).\end{aligned}$$

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

$$\begin{aligned}\mu(A_r) \leq{}& \frac{2\alpha}{(1-\alpha)\pi} e^{-(\alpha-1/2)R-(1-\alpha)r}(1 + \Theta(e^{-\alpha R})) \\ &\cdot \left((1 - e^{-(1-\alpha)(R-r)}) + \Theta\left(e^{R-2r}(1 - e^{-(3-\alpha)(R-r)})\right)\right).\end{aligned}$$

Simplifying the remaining error terms then yields the claim. ◀

We can now bound the maximum clique number in G_ρ^S and thus its interval width $\mathrm{iw}(G_\rho^S)$.

▶ **Theorem 9.** *Let G be a hyperbolic random graph and $r \geq R/2$. Then there exists a constant c such that, whp., $\mathrm{iw}(G_r^S) = \mathcal{O}(\log(n))$ if $r \geq R - \frac{1}{(1-\alpha)} \log\log(n^c)$, and otherwise*

$$\mathrm{iw}(G_r^S) \leq \frac{4\alpha}{(1-\alpha)\pi} n e^{-(\alpha-1/2)R-(1-\alpha)r} \left(1 + \Theta(e^{-\alpha R} + e^{-(2r-R)} - e^{-(1-\alpha)(R-r)})\right).$$

Proof. We start by determining the expected number of arcs that intersect at a given angle, which can be done by computing the expected number of vertices in A_r, using Lemma 8:

$$\mathbb{E}[|\{v \in A_r\}|] \leq \frac{2\alpha}{(1-\alpha)\pi} n e^{-(\alpha-1/2)R-(1-\alpha)r} (1 + \Theta(e^{-\alpha R} + e^{-(2r-R)} - e^{-(1-\alpha)(R-r)})).$$

It remains to show that this bound holds with high probability at every angle. To this end, we make use of a Chernoff bound (Theorem 1), by first showing that the bound on $\mathbb{E}[|\{v \in A_r\}|]$ is $\Omega(\log(n))$. We start with the case where $r < R - \frac{1}{1-\alpha} \log\log(n^c)$.

$$\begin{aligned}
\mathbb{E}[|\{v \in A_r\}|] &< \frac{2\alpha}{(1-\alpha)\pi} n e^{-(\alpha-1/2)R-(1-\alpha)(R-1/(1-\alpha)\log\log(n^c))} \\
&\quad \cdot \Big(1 + \Theta(e^{-\alpha R} + e^{-(2(R-1/(1-\alpha)\log\log(n^c))-R)} \\
&\qquad - e^{-(1-\alpha)(R-(R-1/(1-\alpha)\log\log(n^c)))})\Big) \\
&= \frac{2\alpha}{(1-\alpha)\pi} n e^{-R/2+\log\log(n^c))} \\
&\quad \cdot \Big(1 + \Theta(e^{-\alpha R} + e^{-(R-2/(1-\alpha)\log\log(n^c))} - e^{-\log\log(n^c)})\Big)
\end{aligned}$$

Substituting $R = 2\log(8n/(\pi\bar{\kappa}))$ we obtain

$$\mathbb{E}[|\{v \in A_r\}|] < \frac{\alpha\bar{\kappa}c}{4(1-\alpha)} \log(n)(1 + o(1)).$$

Thus, for all radii smaller than $R - \frac{1}{(1-\alpha)} \log\log(n^c)$, the resulting upper bound is lower bounded by $\Omega(\log(n))$, which lets us apply Theorem 1. Moreover, as $\mathbb{E}[|\{v \in A_r\}|]$ decreases with increasing r, $\mathcal{O}(\log(n))$ is a pessimistic but valid upper bound for the case $r \geq R - \frac{1}{(1-\alpha)} \log\log(n^c)$. Thus, we can also apply Theorem 1 to this case, when using the pessimistic $\mathcal{O}(\log(n))$ bound.

By Theorem 1, we can choose c such that in both cases the bound holds with probability $1 - \mathcal{O}(n^{-c'})$ for any c' at a given angle. In order to see that this also 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 union bound and obtain that the bound holds with probability $1 - \mathcal{O}(n^{-c'+1})$ for any c' at every angle. Since our bound on $\mathbb{E}[|\{v \in A_r\}|]$ is an upper bound on the maximum clique size of G_r^S, it follows that the interval width of G_r^S is at most twice as large, as argued in Section 2. ◀

Since the interval width of a circular arc supergraph of G is an upper bound on the pathwidth of G [8, Theorem 7.14], we immediately obtain the following corollary.

▶ **Corollary 10.** *Let G be a hyperbolic random graph and let G_ρ be the subgraph obtained by removing all vertices with radius at most $\rho = R - 2\log\log(n^c)$. Then,* $\mathrm{pw}(G_\rho) = \mathcal{O}(\log(n))$.

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

▶ **Theorem 2.** *Let G be a hyperbolic random graph on n vertices. Then the* VertexCover *problem in G can be solved in* $\mathrm{poly}(n)$ *time, with high probability.*

Proof. Consider the following algorithm that finds the 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 5, 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_ρ. By Corollary 10, the pathwidth of G_ρ is $\mathcal{O}(\log(n))$, with high probability. Since the pathwidth is an upper bound on the treewidth, we can find a tree decomposition of G_ρ and solve the VertexCover problem in G_ρ in $\mathrm{poly}(n)$ time [8, Theorems 7.18 and 7.14]. ◀

Moreover, linking the radius of a vertex in Theorem 9 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 $G_{\leq d}$. Recall that $G_{\leq d}$ denotes the subgraph of G induced by the vertices of degree at most d.

▶ **Corollary 11.** *Let G be a hyperbolic random graph and let $d \leq \sqrt{n}$. Then, with high probability,* $\mathrm{pw}(G_{\leq d}) = \mathcal{O}(d^{2-2\alpha} + \log(n))$.

Proof. Consider the radius $r = R - 2\log(\varepsilon d)$ for some constant $\varepsilon > 0$, and the graph G_r which is obtained by removing all vertices of radius at most r. By substituting $R = 2\log(8n/(\pi\bar{\kappa}))$ and using [14, Lemma 3.2] we can compute the expected degree of a vertex with radius r as

$$\mathbb{E}[\deg(v) \mid r(v) = r] = \frac{2\alpha}{(\alpha - 1/2)\pi} n e^{-r/2}(1 \pm \mathcal{O}(e^{-(\alpha-1/2)r})) = \frac{\alpha\bar{\kappa}\varepsilon}{4(\alpha - 1/2)} d(1 \pm o(1)).$$

First assume that $d \geq \log(n)^{1/(2-2\alpha)}$. We handle the other case later. Since $d \in \Omega(\log(n))$ we can choose ε large enough to apply Theorem 1 and conclude that this holds with high probability. Furthermore, since a smaller radius implies a larger degree, we know that, with high probability, all nodes v with radius at most r, have

$$\deg(v) \geq \frac{\alpha\bar{\kappa}\varepsilon}{4(\alpha - 1/2)} d(1 \pm o(1)).$$

For large enough n we can choose ε such that, with high probability, G_r is a supergraph of $G_{\leq d}$. To prove the claim, it remains to bound the pathwidth of G_r. If $r > R - 1/(1-\alpha)\log\log(n^c)$, we can apply the first part of Theorem 9 to obtain $\mathrm{iw}(G_r^S) = \mathcal{O}(\log(n))$. Otherwise, we use part two to conclude that the interval width of G_r is at most

$$\begin{aligned}\mathrm{iw}(G_r^S) &\leq \frac{4\alpha}{(1-\alpha)\pi} n e^{-(\alpha-1/2)R-(1-\alpha)r} \left(1 + \Theta(e^{-\alpha R} + e^{-(2r-R)} - e^{-(1-\alpha)(R-r)})\right) \\ &= \frac{\alpha\bar{\kappa}\varepsilon^{2-2\alpha}}{(2-2\alpha)} d^{2-2\alpha} \left(1 + \Theta(n^{-2\alpha} + ((\varepsilon d)^2/n)^2 - (\varepsilon d)^{-(2-2\alpha)})\right) = \mathcal{O}(d^{2-2\alpha}).\end{aligned}$$

As argued in Section 2 the interval width of a graph is an upper bound on the pathwidth.

For $d < \log(n)^{1/(2-2\alpha)}$ (which we excluded above), consider $G_{\leq d'}$ for $d' = \log(n)^{1/(2-2\alpha)} > d$. As we already proved the corollary for d', we obtain $\mathrm{pw}(G_{\leq d'}) = \mathcal{O}(d'^{2-2\alpha} + \log(n)) = \mathcal{O}(\log(n))$. As $G_{\leq d}$ is a subgraph of $G_{\leq d'}$, the same bound holds for $G_{\leq d}$. ◀

4 Discussion

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 run experiments on 59 networks from several network datasets [1, 3, 18, 19, 20]. Although the focus of this paper is the theoretical analysis on hyperbolic random graphs, we briefly report on our experimental results; see Table 1 in Appendix A. Out of the 59 instances, we can solve VertexCover for 47 networks in reasonable time. We refer to these instances 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 5 that all vertices with radius at most $R - 2\log\log(n^{4/\bar{\kappa}})$ probably dominate, which corresponds to an expected degree of $\alpha/(\alpha - 1/2) \cdot \log n$. For more than half of the 59 networks, more than 78 % of the vertices above this degree were in fact dominant. For more than a quarter of the networks, more than 96 % were dominant. Restricted to the 47 easy instances, these number increase to 82 % and 99 %, 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. [21] 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, at most 15 for 34 %, and at most 5 for 25 %. Restricted to easy instances, the values increase to 55 %, 43 %, and 32 %, respectively.

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

Approximation. Concerning approximation algorithms for VertexCover, there is a similar theory-practice gap as for exact solutions. In theory, there is a simple 2-approximation and the best known polynomial time approximation reduces the factor to $2 - \Theta(\log(n)^{-1/2})$ [15]. However, it is NP-hard to approximate VertexCover within a factor of 1.3606 [10], and presumably it is even NP-hard to approximate within a factor of $2 - \varepsilon$ for all $\varepsilon > 0$ [16]. Moreover, the greedy strategy that iteratively adds the vertex with maximum degree to the vertex cover and deletes it, is only a $\log n$ approximation. However, on scale-free networks this strategy performs exceptionally well with approximation ratios very close to 1 [9].

Our results for hyperbolic random graphs at least partially explain this good approximation ratio. Lemma 5 states that, with high probability, we do not make any mistake by taking all vertices below a certain radius ρ, which corresponds to vertices of at least logarithmic degree. The same computation for larger values of ρ does no longer give such strong guarantees. However, it still gives bounds on the probability for making a mistake. In fact, this error probability is sub-constant as long as the corresponding expected degree is super-constant.

Although this is not a formal argument, it still explains to a degree why greedy works so well on networks with a heterogeneous degree distribution and high clustering. Moreover, it indicates how the greedy algorithm should be adapted to obtain even better approximation ratios: As the probability to make a mistake grows with growing radius and thus with shrinking vertex degree, the majority of mistakes are done when all vertices have already low degree. However, for hyperbolic random graphs, the subgraphs induced by vertices below a certain constant degree decompose into small components for $n \to \infty$. It thus seems to be a good idea to run the greedy algorithm only until all remaining vertices have low degree, say k. The remaining small connected components of maximum-degree k can then be solved with brute force. In the following we call the resulting algorithm *k-adaptive greedy*.

We ran experiments on the 47 easy real networks mentioned above (for the hard instances, we cannot measure approximation ratios). For these networks, we compare the normal greedy algorithm with 2- and 4-adaptive greedy. Note that 2-adaptive greedy is special, as VertexCover can be solved efficiently on graphs with maximum degree 2 (no brute-forcing is necessary). For 4-adaptive greedy, the size of the largest connected component is relevant.

The median approximation ratio for greedy over all 47 networks is 1.008. This goes down to 1.005 for 2-adaptive and to 1.002 for 4-adaptive greedy. Thus, the number of too many selected vertices goes down by a factor of 1.6 and 4, respectively. As mentioned above, the size of the largest connected component is relevant for 4-adaptive greedy. For 49 % of the networks, this was below 100 (which is still a reasonable size for a brute-force algorithm). Restricted to these networks, normal greedy has a median approximation ratio of 1.004, while 4-adaptive again improves by a factor of 4 to 1.001. Moreover, the number of networks for which we actually obtain the optimal solution increases from 4 to 7.

References

1. Gephi datasets. `https://github.com/gephi/gephi/wiki/Datasets`.
2. Takuya Akiba and Yoichi Iwata. Branch-and-reduce exponential/FPT algorithms in practice: A case study of vertex cover. *Theoretical Computer Science*, 609:211 – 225, 2016. doi:10.1016/j.tcs.2015.09.023.
3. Vladimir Batagelj and Andrej Mrvar. Pajek datasets. `http://vlado.fmf.uni-lj.si/pub/networks/data/`, 2006.
4. Thomas Bläsius, Tobias Friedrich, and Anton Krohmer. Hyperbolic Random Graphs: Separators and Treewidth. In *24th Annual European Symposium on Algorithms (ESA 2016)*, pages 15:1–15:16, 2016. doi:10.4230/LIPIcs.ESA.2016.15.
5. Marián Boguná, Fragkiskos Papadopoulos, and Dmitri Krioukov. Sustaining the internet with hyperbolic mapping. *Nature Communications*, 1:62, 2010. doi:10.1038/ncomms1063.
6. Karl Bringmann, Ralph Keusch, Johannes Lengler, Yannic Maus, and Anisur Rahaman Molla. Greedy routing and the algorithmic small-world phenomenon. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '17, pages 371–380. ACM, 2017. doi:10.1145/3087801.3087829.
7. Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40):3736 – 3756, 2010. doi:10.1016/j.tcs.2010.06.026.
8. Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
9. 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):1350078, 2013. doi:10.1142/S0129183113500782.
10. Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439 – 485, 2005. doi:10.4007/annals.2005.162.439.
11. Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2012.
12. Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. A measure & conquer approach for the analysis of exact algorithms. *J. ACM*, 56(5):25:1 – 25:32, August 2009. doi:10.1145/1552285.1552286.

13 Tobias Friedrich and Anton Krohmer. On the diameter of hyperbolic random graphs. In *Automata, Languages, and Programming*, pages 614 – 625. Springer Berlin Heidelberg, 2015. doi:10.1007/978-3-662-47666-6_49.

14 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.

15 George Karakostas. A better approximation ratio for the vertex cover problem. *ACM Trans. Algorithms*, 5(4):41:1–41:8, November 2009. doi:10.1145/1597036.1597045.

16 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2 - \varepsilon$. *Journal of Computer and System Sciences*, 74(3):335 – 349, 2008. doi:10.1016/j.jcss.2007.06.019.

17 Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, Sep 2010. doi:10.1103/PhysRevE.82.036106.

18 Jérôme Kunegis. KONECT: The koblenz network collection. In *International Conference on World Wide Web (WWW)*, pages 1343–1350, 2013. doi:10.1145/2487788.2488173.

19 Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, 2014.

20 Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. URL: http://networkrepository.com.

21 Hisao Tamaki, Hiromu Ohtsuka, Takuto Sato, and Keitaro Makii. Tcs-meiji pace2017-tracka. github.com/TCS-Meiji/PACE2017-TrackA, 2017.

22 Mingyu Xiao and Hiroshi Nagamochi. Exact algorithms for maximum independent set. *Information and Computation*, 255:126 – 146, 2017. doi:10.1016/j.ic.2017.06.001.

A Experimental Data

Table 1 (continuing on the next page) shows the raw data of our experiments for which we reported aggregate values in the discussion in Section 4. The percentage of dominant vertices among those with high degree (over $\alpha/(\alpha - 1/2) \cdot \log n$) is rounded to whole percentages. The approximation ratios are rounded to three decimal digits. Treewidth -1 indicates that remaining graph after removing all dominant vertices contained no edge.

Table 1 The resulting raw data of our experiments. The columns are: **(network)** the network name; **(easy)** whether or not the network is easy; **(dom)**the percentage of dominant nodes among those of degree above the threshold $\alpha/(\alpha - 1/2) \cdot \log n$; **(tw)**an upper bound for the treewidth of the remaining graph after deleting dominant nodes; **(greedy)** the approximation ratio of greedy; **(2-ad)** of 2-adaptive greedy; **(4-ad)** of 4-adaptive greedy; **(comp)** the size of the largest component that remains after the greedy phase of 4-adaptive greedy.

network	easy	dom	tw	greedy	2-ad	4-ad	comp
advogato	✓	51 %	314	1.011	1.009	1.005	863
airlines	✓	28 %	23	1.000	1.000	1.000	75
as-22july06	✓	100 %	3	1.002	1.001	1.001	46
as-caida20071105	✓	100 %	3	1.002	1.001	1.000	35
as-skitter	✗	47 %	969794				
as20000102	✓	100 %	2	1.003	1.001	1.001	18
bio-CE-HT	✓	100 %	3	1.015	1.009	1.000	225
bio-CE-LC	✓	100 %	2	1.003	1.003	1.003	39
bio-DM-HT	✓	50 %	13	1.017	1.014	1.004	319
bio-yeast-protein-inter	✓	100 %	4	1.013	1.006	1.002	147
bn-fly-drosophila-medulla-1	✓	72 %	38	1.018	1.013	1.009	142
bn-mouse-kasthuri-graph-v4	✓	100 %	1	1.006	1.000	1.000	12
ca-AstroPh	✓	94 %	6	1.003	1.002	1.001	123
ca-cit-HepPh	✓	84 %	151	1.003	1.003	1.002	533
ca-CondMat	✓	99 %	4	1.003	1.002	1.001	53
ca-GrQc	✓	99 %	2	1.004	1.002	1.001	44
ca-HepTh	✓	95 %	13	1.005	1.004	1.001	174
cfinder-google	✗	66 %	82				
cit-HepTh	✗	13 %	19737				
citeseer	✗	46 %	182372				
com-amazon	✓	93 %	2756	1.011	1.006	1.002	16209
com-dblp	✓	100 %	7	1.002	1.001	1.000	69
cpan-authors	✓	100 %	2	1.009	1.009	1.009	17
digg-friends	✓	58 %	1649	1.008	1.006	1.004	179
ego-facebook	✓	100 %	-1	1.000	1.000	1.000	3
ego-gplus	✓	100 %	1	1.000	1.000	1.000	5
email-Enron	✓	85 %	41	1.003	1.002	1.001	141
EuroSiS	✓	56 %	34	1.020	1.018	1.010	274
facebook-wosn-links	✗	27 %	36694				
flixster	✗	73 %	122				
hyves	✓	98 %	1653	1.008	1.008	1.008	42
livemocha	✓	4 %	24380	1.017	1.013	1.006	25300

Table 1 The resulting raw data of our experiments. The columns are: **(network)** the network name; **(easy)** whether or not the network is easy; **(dom)**the percentage of dominant nodes among those of degree above the threshold $\alpha/(\alpha - 1/2) \cdot \log n$; **(tw)**an upper bound for the treewidth of the remaining graph after deleting dominant nodes; **(greedy)** the approximation ratio of greedy; **(2-ad)** of 2-adaptive greedy; **(4-ad)** of 4-adaptive greedy; **(comp)** the size of the largest component that remains after the greedy phase of 4-adaptive greedy.

network	easy	dom	tw	greedy	2-ad	4-ad	comp
loc-brightkite-edges	✓	76 %	619	1.014	1.009	1.004	4658
loc-gowalla-edges	✗	64 %	3991				
moreno-names	✓	94 %	3	1.006	1.004	1.002	34
moreno-propro	✓	100 %	4	1.014	1.006	1.002	153
munmun-twitter-social	✓	57 %	12	1.000	1.000	1.000	5
OClinks	✓	36 %	202	1.017	1.015	1.005	498
p2p-Gnutella04	✓	42 %	1352	1.019	1.017	1.016	970
p2p-Gnutella05	✓	40 %	1075	1.014	1.013	1.013	447
p2p-Gnutella06	✓	40 %	1142	1.023	1.022	1.021	820
p2p-Gnutella08	✓	47 %	414	1.008	1.008	1.008	45
p2p-Gnutella09	✓	47 %	419	1.005	1.005	1.005	63
p2p-Gnutella24	✓	81 %	525	1.006	1.005	1.005	70
p2p-Gnutella25	✓	79 %	464	1.006	1.005	1.005	77
p2p-Gnutella30	✓	79 %	604	1.005	1.005	1.004	62
p2p-Gnutella31	✓	80 %	732	1.011	1.010	1.010	65
petster-carnivore	✓	79 %	149312	1.008	1.007	1.004	9238
petster-friendship-cat	✗	12 %	14929				
petster-friendship-dog	✗	15 %	340634				
petster-friendship-hamster	✗	23 %	135				
soc-Epinions1	✓	82 %	238	1.006	1.003	1.001	228
US-Air	✓	67 %	4	1.013	1.000	1.000	23
web-Google	✗	84 %	103939				
wiki-Vote	✓	44 %	384	1.054	1.052	1.050	726
wordnet-words	✓	95 %	28	1.004	1.003	1.002	59
YeastS	✓	70 %	39	1.013	1.012	1.005	244
youtube-links	✓	86 %	1239	1.008	1.004	1.001	570
youtube-u-growth	✗	90 %	59358				