

# Sampling Geometric Inhomogeneous Random Graphs in Linear Time\*

Karl Bringmann<sup>1</sup>, Ralph Keusch<sup>2</sup>, and Johannes Lengler<sup>3</sup>

1 Max-Planck-Institute for Informatics, Saarbrücken, Germany  
kbringma@mpi-inf.mpg.de

2 Institute of Theoretical Computer Science, ETH Zürich, Switzerland  
rkeusch@inf.ethz.ch

3 Institute of Theoretical Computer Science, ETH Zürich, Switzerland  
lenglerj@inf.ethz.ch

---

## Abstract

Real-world networks, like social networks or the internet infrastructure, have structural properties such as large clustering coefficients that can best be described in terms of an underlying geometry. This is why the focus of the literature on theoretical models for real-world networks shifted from classic models without geometry, such as Chung-Lu random graphs, to modern geometry-based models, such as hyperbolic random graphs.

With this paper we contribute to the theoretical analysis of these modern, more realistic random graph models. Instead of studying directly hyperbolic random graphs, we introduce a generalization that we call *geometric inhomogeneous random graphs* (GIRGs). Since we ignore constant factors in the edge probabilities, GIRGs are technically simpler (specifically, we avoid hyperbolic cosines), while preserving the qualitative behaviour of hyperbolic random graphs, and we suggest to replace hyperbolic random graphs by this new model in future theoretical studies.

We prove the following fundamental structural and algorithmic results on GIRGs. (1) As our main contribution we provide a sampling algorithm that generates a random graph from our model in expected *linear* time, improving the best-known sampling algorithm for hyperbolic random graphs by a substantial factor  $O(\sqrt{n})$ . (2) We establish that GIRGs have clustering coefficients in  $\Omega(1)$ , (3) we prove that GIRGs have small separators, i.e., it suffices to delete a sublinear number of edges to break the giant component into two large pieces, and (4) we show how to compress GIRGs using an expected linear number of bits.

**1998 ACM Subject Classification** G.2.2 Graph Theory, F.2.2 Nonnumerical Algorithms and Problems

**Keywords and phrases** real-world networks, random graph models, sampling algorithms, compression algorithms, hyperbolic random graphs

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2017.20

## 1 Introduction

Real-world networks, like social networks or the internet infrastructure, have structural properties that can best be described using *geometry*. For instance, in social networks two people are more likely to know each other if they live in the same region and share hobbies, both of which can be encoded as spatial information. This geometric structure may be responsible for some of the key properties of real-world networks, e.g., an underlying geometry

---

\* A full version of this paper, including all proofs, is available at <http://arxiv.org/abs/1511.00576>.



naturally induces a large number of triangles, or large *clustering coefficient*: Two of one’s friends are likely to live in one’s region and have similar hobbies, so they are themselves similar and thus likely to know each other.

Classic mathematical models of real-world networks are *scale-free* (i.e., have a power-law degree distribution) and *small worlds* (i.e., most pairs of vertices have small graph-theoretic distance), thus reproducing these two key findings of large real-world networks. But since they have no underlying geometry their clustering coefficient is as small as  $n^{-\Omega(1)}$ ; this holds in particular for preferential attachment graphs [3] and Chung-Lu random graphs [24, 25, 26] (and their variants [13, 41]). In order to close this gap between the empirically observed clustering coefficient and theoretical models, much of the recent work on models for real-world networks focussed on scale-free random graph models that are equipped with an underlying geometry, such as hyperbolic random graphs [11, 42], spatial preferred attachment [2], and many others [13, 14, 15, 35]. The basic properties – scale-freeness, small-world, and large clustering coefficient – have been rigorously established for most of these models. Beyond the basics, experiments suggest that these models have some very desirable properties.

In particular, hyperbolic random graphs are a promising model, as Boguñá et al. [11] computed a (heuristic) maximum likelihood fit of the internet graph into the hyperbolic random graph model and demonstrated its quality by showing that greedy routing in the underlying geometry of the fit finds near-optimal shortest paths. Further properties that have been studied on hyperbolic random graphs, mostly agreeing with empirical findings on real-world networks, are scale-freeness and clustering coefficient [33, 20], existence of a giant component [9], diameter [37, 32], average distance [1], separators and treewidth [6], spectral gap [38], bootstrap percolation [21], and clique number [7]. Algorithmic aspects include sampling algorithms [47], embedding algorithms [8], and compression schemes [45].

Our goal is to improve algorithmic and structural results on the promising model of hyperbolic random graphs. However, it turns out to be beneficial to work with a more general model, that we introduce with this paper: In a *geometric inhomogeneous random graph* (GIRG), every vertex  $v$  comes with a weight  $w_v$  (which we assume to follow a power law in this paper) and picks a uniformly random position  $x_v$  in the  $d$ -dimensional torus  $\mathbb{T}^d$ . Two vertices  $u, v$  then form an edge independently with probability  $p_{uv}$ , which is proportional to  $w_u w_v$  and inversely proportional to some power of their distance  $\|x_u - x_v\|$ , see Section 2 for details. A major difference between hyperbolic random graphs and our generalisation is that we ignore constant factors in the edge probabilities  $p_{uv}$ . This allows to greatly simplify the edge probability expressions, thus reducing the technical overhead. GIRGs can be interpreted as a geometric variant of the classic Chung-Lu random graphs. Recently, with scale-free percolation a closely related model has been introduced [28] where the vertex set is given by the grid  $\mathbb{Z}^d$ . This model is similar with respect to component structure, clustering, and small-world properties [29, 34], but none of the algorithmic aspects studied in the present paper (sampling, compression, also separators) has been regarded thereon.

The basic connectivity properties of GIRGs follow from more general considerations in [17], where an even more general model of generic augmented Chung-Lu graphs is studied. In particular, with high probability<sup>1</sup> GIRGs have a giant component and polylogarithmic diameter, and a.a.s. doubly-logarithmic average distance within the giant. However, general studies such as [17] are limited to properties that do not depend on the specific underlying geometry. Recently, GIRGs turned out to be accesible for studying processes such as bootstrap percolation [39] and greedy routing [19].

---

<sup>1</sup> We say that an event holds *with high probability* (w.h.p.) if it holds with probability  $1 - n^{-\omega(1)}$ . If it holds with probability  $1 - o(1)$ , we say that it holds *asymptotically almost surely* (a.a.s.).

**Our contribution.** As our main result, we present a sampling algorithm that generates a random graph from our model in expected linear time. This improves the trivial sampling algorithm by a factor  $O(n)$  and the best-known algorithm for hyperbolic random graphs by a factor  $O(\sqrt{n})$  [47]. We also prove that the underlying geometry indeed causes GIRGs to have a clustering coefficient in  $\Omega(1)$ . Moreover, we show that GIRGs have small separators of expected size  $n^{1-\Omega(1)}$ ; this is in agreement with empirical findings on real-world networks [5]. We then use the small separators to prove that GIRGs can be efficiently compressed (i.e., they have low entropy), specifically, we show how to store a GIRG using  $O(n)$  bits in expectation. Finally, we show that hyperbolic random graphs are indeed a special case of GIRGs, so that all aforementioned results also hold for hyperbolic random graphs.

## 2 Model and Results

### 2.1 Definition of the Model

We prove algorithmic and structural results in a new random graph model which we call *geometric inhomogeneous random graphs*. In this model, each vertex  $v$  comes with a weight  $w_v$  and with a random position  $x_v$  in a geometric space, and the set of edges  $E$  is also random. We start by defining the by-now classical Chung-Lu model and then describe the changes that yield our variant with underlying geometry.

**Chung-Lu random graph.** For  $n \in \mathbb{N}$  let  $w = (w_1, \dots, w_n)$  be a sequence of positive weights. We call  $W := \sum_{v=1}^n w_v$  the *total weight*. The Chung-Lu random graph  $G(n, w)$  has vertex set  $V = [n] = \{1, \dots, n\}$ , and two vertices  $u \neq v$  are connected by an edge independently with probability  $p_{uv} = \Theta(\min\{1, \frac{w_u w_v}{W}\})$  [24, 25]. Note that the term  $\min\{1, \cdot\}$  is necessary, as the product  $w_u w_v$  may be larger than  $W$ . Classically, the  $\Theta$  simply hides a factor 1, but by introducing the  $\Theta$  the model also captures similar random graphs, like the Norros-Reittu model [41], while important properties stay asymptotically invariant.

**Geometric inhomogeneous random graph (GIRG).** Note that we obtain a circle by identifying the endpoints of the interval  $[0, 1]$ . Then the distance of  $x, y \in [0, 1]$  along the circle is  $|x - y|_C := \min\{|x - y|, 1 - |x - y|\}$ . We fix a dimension  $d \geq 1$  and use as our *ground space* the  $d$ -dimensional torus  $\mathbb{T}^d = \mathbb{R}^d / \mathbb{Z}^d$ , which can be described as the  $d$ -dimensional cube  $[0, 1]^d$  where opposite boundaries are identified. As distance function we use the  $\infty$ -norm on  $\mathbb{T}^d$ , i.e., for  $x, y \in \mathbb{T}^d$  we define  $\|x - y\| := \max_{1 \leq i \leq d} |x_i - y_i|_C$ .

As for Chung-Lu graphs, we consider the vertex set  $V = [n]$  and a weight sequence  $w$  (in this paper we require the weights to follow a power law with exponent  $\beta > 2$ , see next paragraph). Additionally, for any vertex  $v$  we draw a point  $x_v \in \mathbb{T}^d$  uniformly and independently at random. Again we connect vertices  $u \neq v$  independently with probability  $p_{uv} = p_{uv}(r)$ , which now depends not only on the weights  $w_u, w_v$  but also on the positions  $x_u, x_v$ , more precisely, on the distance  $r = \|x_u - x_v\|$ . We require for some constant  $\alpha > 1$  the following edge probability condition:

$$p_{uv} = \Theta\left(\min\left\{\frac{1}{\|x_u - x_v\|^{\alpha d}} \cdot \left(\frac{w_u w_v}{W}\right)^\alpha, 1\right\}\right). \quad (\text{EP1})$$

We also allow  $\alpha = \infty$  and in this case require that

$$p_{uv} = \begin{cases} \Theta(1) & \text{if } \|x_u - x_v\| \leq O\left(\left(\frac{w_u w_v}{W}\right)^{1/d}\right) \\ 0 & \text{if } \|x_u - x_v\| \geq \Omega\left(\left(\frac{w_u w_v}{W}\right)^{1/d}\right), \end{cases} \quad (\text{EP2})$$

where the constants hidden by  $O$  and  $\Omega$  do not have to match, i.e., there can be an interval  $[c_1(\frac{w_u w_v}{W})^{1/d}, c_2(\frac{w_u w_v}{W})^{1/d}]$  for  $\|x_u - x_v\|$  where the behaviour of  $p_{uv}$  is arbitrary. This finishes the definition of GIRGs. The free parameters of the model are  $\alpha \in (1, \infty]$ ,  $d \in \mathbb{N}$ , the concrete weights  $w$  with power-law exponent  $\beta > 2$  and average weight  $W/n$ , the concrete function  $f_{uv}(x_u, x_v)$  replacing the  $\Theta$  in  $p_{uv}$ , and for  $\alpha = \infty$  the constants hidden by  $O, \Omega$  in the requirement for  $p_{uv}$ . We will typically hide the constants  $\alpha, d, \beta, W/n$  by  $O$ -notation.

**Power-law weights.** As is often done for Chung-Lu graphs, we assume throughout this paper that the weights follow a *power law*: the fraction of vertices with weight at least  $w$  is proportional to  $w^{1-\beta}$  for some  $2 < \beta < 3$  (the *power-law exponent* of  $w$ ). More precisely, we assume that for some  $\bar{w} = \bar{w}(n)$  with  $n^{\omega(1/\log \log n)} \leq \bar{w} \leq n^{(1-\Omega(1))/(\beta-1)}$ , the sequence  $w$  satisfies the following conditions:

(PL1) the minimum weight is constant, i.e.,  $w_{\min} := \min\{w_v \mid 1 \leq v \leq n\} = \Omega(1)$ ;

(PL2) for all  $\eta > 0$  there exist constants  $c_1, c_2 > 0$  such that

$$c_1 \frac{n}{w^{\beta-1+\eta}} \leq \#\{1 \leq v \leq n \mid w_v \geq w\} \leq c_2 \frac{n}{w^{\beta-1-\eta}},$$

where the first inequality holds for all  $w_{\min} \leq w \leq \bar{w}$  and the second for all  $w \geq w_{\min}$ . We remark that these are standard assumptions for power-law graphs with average degree  $\Theta(1)$ . In particular, (PL2) implies that the average weight  $W/n$  is  $\Theta(1)$ . An example is the widely used weight function  $w_v := \delta \cdot (n/v)^{1/(\beta-1)}$  with parameter  $\delta = \Theta(1)$ .

**Discussion of the model.** The choice of the ground space  $\mathbb{T}^d$  is in the spirit of the classic random geometric graphs [44]. We prefer the torus to the hyper-cube for technical simplicity, as it yields symmetry. However, one could replace  $\mathbb{T}^d$  by  $[0, 1]^d$  or other manifolds like the  $d$ -dimensional sphere; our results will still hold verbatim. Moreover, since in fixed dimension all  $L_p$ -norms on  $\mathbb{T}^d$  are equivalent and since the edge probabilities  $p_{uv}$  have a constant factor slack, our choice of the  $L_\infty$ -norm is without loss of generality (among all norms).

The model is already motivated since it generalizes the celebrated hyperbolic random graphs (see Theorem 7). Let us nevertheless discuss why our choice of edge probabilities is *natural*: The term  $\min\{., 1\}$  is necessary, as in the Chung-Lu model, because  $p_{uv}$  is a probability. To obtain a geometric model, where adjacent vertices are likely to have small distance,  $p_{uv}$  should decrease with increasing distance  $\|x_u - x_v\|$ , and an inverse polynomial relation seems reasonable. The constraint  $\alpha > 1$  is necessary to cancel the growth of the volume of the ball of radius  $r$  proportional to  $r^d$ , so that we expect most neighbors of a vertex to lie close to it. Finally, the factor  $(\frac{w_u w_v}{W})^\alpha$  ensures that the marginal probability of vertices  $u, v$  with weights  $w_u, w_v$  forming an edge is  $\Pr[u \sim v] = \Theta(\min\{\frac{w_u w_v}{W}, 1\})$ , as in the Chung-Lu model, and this probability does not change by more than a constant factor if we fix either  $x_u$  or  $x_v$ . This is why we see our model as a geometric variant of Chung-Lu random graphs. For a fixed vertex  $u \in V$  we can sum up  $\Pr[u \sim v \mid x_u]$  over all vertices  $v \in V \setminus \{u\}$ , and it follows  $\mathbb{E}[\deg(u)] = \Theta(w_u)$ . The main reason why GIRGs are also *technically easy* is that for any vertex  $u$  with fixed position  $x_u$  the incident edges  $\{u, v\}$  are independent.

Finally, after rescaling the parameters ( $\tilde{x}_v := n^{1/d} x_v$ ,  $\tilde{\alpha} := d\alpha$ ,  $\tau := 1 + (\beta - 1)/\alpha$ , see [46] for details), the GIRG model is closely related to scale-free percolation [28].

**Sampling the weights.** In the definition we assume that the weight sequence  $w$  is fixed. However, if we sample the weights independently according to an appropriate power-law distribution with minimum weight  $w_{\min}$  and density  $f(w) \sim w^{-\beta}$ , then the sampled weight sequence will follow a power law and fulfils (PL1) and (PL2) with probability  $1 - n^{-\Omega(1)}$ . Hence, a model with sampled weights is a.a.s. included in our model.

## 2.2 Structural Properties of GIRGs

As discussed in the introduction, reasonable random graph models for real-world networks should reproduce a power-law degree distribution and small graph-theoretical distances between nodes. For the GIRG model, these structural properties follow from a more general class of generic augmented Chung-Lu random graphs that have been studied in [17]. This framework has weaker assumptions on the underlying geometry than GIRGs. A short comparison reveals that GIRGs are a special case of this general class of random graph models. In the following we list the results of [17] transferred to GIRGs. As we are using power-law weights and  $\mathbb{E}[\deg(v)] = \Theta(w_v)$  holds for all  $v \in V$ , it is not surprising that the degree sequence follows a power-law.

► **Theorem 1** (Theorem 2.1 in [17]). *W.h.p. the degree sequence of a GIRG follows a power law with exponent  $\beta$  and average degree  $\Theta(1)$ .*

The next result determines basic connectivity properties. Note that for  $\beta > 3$ , they are not well-behaved, in particular since in this case even threshold hyperbolic random graphs do not possess a giant component of linear size [10]. This is one reason why we assume  $2 < \beta < 3$  throughout the paper. For the following theorem, we require the additional assumption  $\bar{w} = \omega(n^{1/2})$  in the limit case  $\alpha = \infty$ .

► **Theorem 2** (Theorems 2.2 and 2.3 in [17]). *W.h.p. the largest component of a GIRG has linear size and diameter  $\log^{O(1)} n$ , while all other components have size  $\log^{O(1)} n$ . Moreover, the average distance of vertices in the largest component is  $(2 \pm o(1)) \frac{\log \log n}{|\log(\beta-2)|}$  in expectation and with probability  $1 - o(1)$ .*

We remark that most results of this paper crucially depend on an underlying geometry, and thus do *not* hold in the general model from [17].

## 2.3 Results

**Sampling.** Sampling algorithms that generate a random graph from a fixed distribution are known for Chung-Lu random graphs and others, running in expected linear time [4, 40]. As our main result, we present such an algorithm for GIRGs. This greatly improves the trivial  $O(n^2)$  sampling algorithm (throwing a biased coin for each possible edge), as well as the best previous algorithm for threshold hyperbolic random graphs with expected time  $O(n^{3/2})$  [47]. It allows to run experiments on much larger graphs than the ones with  $\approx 10^4$  vertices in [11]. In addition to our model assumptions, here we assume that the  $\Theta$  in our requirement on  $p_{uv}$  is sufficiently explicit, i.e., we can compute  $p_{uv}$  exactly and we know a constant  $c > 0$  such that replacing  $\Theta$  by  $c$  yields an upper bound on  $p_{uv}$ , see Section 3 for details.

► **Theorem 3** (Section 3). *Sampling a GIRG can be done in expected time  $O(n)$ .*

**Clustering.** In social networks, two friends of the same person are likely to also be friends with each other. This property of having many triangles is captured by the clustering coefficient, defined as the probability when choosing a random vertex  $v$  and two random neighbors  $v_1 \neq v_2$  of  $v$  that  $v_1$  and  $v_2$  are adjacent (if  $v$  does not have two neighbors then its contribution to the clustering coefficient is 0). While Chung-Lu random graphs have a very small clustering coefficient of  $n^{-\Omega(1)}$ , it is easy to show that the clustering coefficient of GIRGs is  $\Theta(1)$ . This is consistent with empirical data of real-world networks [31] and the constant clustering coefficient of hyperbolic random graphs determined in [20, 33, 45].

► **Theorem 4.** *W.h.p. the clustering coefficient of a GIRG is  $\Theta(1)$ .*

**Proof Outline.** We show that the clustering coefficient is dominated by the contribution of constant-weight vertices  $v$ . Let  $v \in V$  be a vertex of weight  $w_v = \Theta(1)$ . Then, with at least constant probability, (i)  $\deg(v) \geq 2$ , and (ii) all neighbors of  $v$  are located in a ball of radius  $cn^{-1/d}$  around  $x_v$ , for a sufficiently small constant  $c > 0$ . If the neighborhood of  $v$  has this property, then two random neighbors  $v_1, v_2$  of  $v$  are connected with constant probability. Therefore, the expected contribution of  $v$  to the clustering coefficient is  $\Omega(\frac{1}{n})$ . As the number of such vertices  $v$  is  $\Theta(n)$ , it follows that the expected clustering coefficient is  $\Theta(1)$ . Proving the w.h.p.-statement requires additional arguments and the application of Azuma-type concentration inequalities with bad events. The detailed proof is included in the full version [18]. ◀

**Stability.** For real-world networks, a key property to analyze is their stability under attacks. It has been empirically observed that many real-world networks have small separators of size  $n^c$ ,  $c < 1$  [5]. In contrast, Chung-Lu random graphs are unrealistically stable, since any deletion of  $o(n)$  nodes or edges reduces the size of the giant component by at most  $o(n)$  [13]. We show that GIRGs agree with the empirical results much better. Specifically, if we cut the ground space  $\mathbb{T}^d$  into two halves along one of the axes then we roughly split the giant component into two halves, but the number of edges passing this cut is quite small, namely  $n^{1-\Omega(1)}$ . Thus, GIRGs are prone to (quite strong) adversarial attacks, just as many real-world networks. Furthermore, their small separators are useful for many algorithms, e.g., the compression scheme of the next paragraph.

► **Theorem 5** (Section 4). *A.a.s. it suffices to delete  $O(n^{\max\{2-\alpha, 3-\beta, 1-1/d\}+o(1)})$  edges of a GIRG to split its giant component into two parts of linear size each.*

Since we assume  $\alpha > 1$ ,  $\beta > 2$ , and  $d = \Theta(1)$ , the number of deleted edges is indeed  $n^{1-\Omega(1)}$ . Recently, Bläsius et al. [6] proved a better bound of  $O(n^{(3-\beta)/2})$  for threshold hyperbolic random graphs which correspond to GIRGs with parameters  $d = 1$  and  $\alpha = \infty$ .

**Entropy.** The internet graph has empirically been shown to be well compressible, using only 2-3 bits per edge [5, 12]. This is not the case for the Chung-Lu model, as its entropy is  $\Theta(n \log n)$  [23]. We show that GIRGs have linear entropy, as is known for threshold hyperbolic random graphs [45].

► **Theorem 6** (Section 4). *We can store a GIRG using  $O(n)$  bits in expectation. The resulting data structure allows to query the degree of any vertex and its  $i$ -th neighbor in time  $O(1)$ . The compression algorithm runs in time  $O(n)$ .*

**Hyperbolic random graphs.** We establish that hyperbolic random graphs are an example of one-dimensional GIRGs, and that the often studied special case of threshold hyperbolic graphs is obtained by our limit case  $\alpha = \infty$ . Specifically, we obtain hyperbolic random graphs from GIRGs by setting the dimension  $d = 1$ , the weights to a specific power law, and the  $\Theta$  in the edge probability  $p_{uv}$  to a specific, complicated function.

► **Theorem 7.** *For every choice of parameters in the hyperbolic random graph model, there is a choice of parameters in the GIRG model such that the two resulting distributions of graphs coincide.*

In particular, all our results on GIRGs hold for hyperbolic random graphs, too. Moreover, as our proofs are much less technical than typical proofs for hyperbolic random graphs, we suggest to switch from hyperbolic random graphs to GIRGs in future studies. We prove Theorem 7 in the full version of this paper [18].

## 2.4 Preliminaries

We introduce a geometric ordering of the vertices, which we will use both for the sampling and for the compression algorithm. Consider the ground space  $\mathbb{T}^d$ , split it into  $2^d$  equal cubes, and repeat this process with each created cube; we call the resulting cubes *cells*. Cells are cubes of the form  $C = [x_1 2^{-\ell}, (x_1 + 1) 2^{-\ell}] \times \dots \times [x_d 2^{-\ell}, (x_d + 1) 2^{-\ell}]$  with  $\ell \geq 0$  and  $0 \leq x_i < 2^\ell$ . We represent cell  $C$  by the tuple  $(\ell, x_1, \dots, x_d)$ . The volume of  $C$  is  $\text{VOL}(C) = 2^{-\ell \cdot d}$ . For  $0 < x \leq 1$  we let  $\lceil x \rceil_{2^d}$  be the smallest number larger or equal to  $x$  that is realized as the volume of a cell, or in other words  $x$  rounded up to a power of  $2^d$ ,  $\lceil x \rceil_{2^d} = \min\{2^{-\ell \cdot d} \mid \ell \in \mathbb{N}_0: 2^{-\ell \cdot d} \geq x\}$ . Note that the cells of a fixed level  $\ell$  partition the ground space. We obtain a *geometric ordering* of these cells by following the recursive construction of cells in a breadth-first-search manner. This yields the following lemma.

► **Lemma 8** (Geometric ordering). *There is an enumeration of the cells  $C_1, \dots, C_{2^{\ell d}}$  of level  $\ell$  such that for every cell  $C$  of level  $\ell' < \ell$  the cells of level  $\ell$  contained in  $C$  form a consecutive block  $C_i, \dots, C_j$  in the enumeration.*

## 3 Sampling Algorithm

In this section we show that GIRGs can be sampled in expected time  $O(n)$ . The running time depends exponentially on the fixed dimension  $d$ . In addition to our model assumptions, in this section we require that (1) edge probabilities  $p_{uv}$  can be computed in constant time (given any vertices  $u, v$  and positions  $x_u, x_v$ ) and (2) we know an explicit constant  $c > 0$  such that if  $\alpha < \infty$  we have

$$p_{uv} \leq \min \left\{ c \frac{1}{\|x_u - x_v\|^{\alpha d}} \cdot \left( \frac{w_u w_v}{W} \right)^\alpha, 1 \right\}.$$

Note that existence of  $c$  follows from our model assumptions. In the remainder of this section we introduce building blocks of our algorithm (Section 3.1) and present our algorithm (Section 3.2) and its analysis (Section 3.3). Note that in the full version, we also show how the sampling algorithm can be adapted to the case  $\alpha = \infty$ .

### 3.1 Building Blocks

**Data structures.** We first build a basic data structure on a set of points  $P$  that allows to access the points in a given cell  $C$  (of volume at least  $\nu$ ) in constant time.

► **Lemma 9.** *Given a set of points  $P$  and  $0 < \nu \leq 1$ , in time  $O(|P| + 1/\nu)$  we can construct a data structure  $\mathcal{D}_\nu(P)$  supporting the following queries in time  $O(1)$ :*

- *given a cell  $C$  of volume at least  $\nu$ , return  $|C \cap P|$ ,*
- *given a cell  $C$  of volume at least  $\nu$  and a number  $k$ , return the  $k$ -th point in  $C \cap P$  (in a fixed ordering of  $C \cap P$  depending only on  $P$  and  $\nu$ ).*

**Proof.** Let  $\mu = \lceil \nu \rceil_{2^d} = 2^{-\ell \cdot d}$ , so that  $\nu \leq \mu \leq O(\nu)$ . Following the recursive construction of cells, we can determine a geometric ordering of the cells of volume  $\mu$  as in Lemma 8 in time  $O(1/\mu) = O(1/\nu)$ ; say  $C_1, \dots, C_{1/\mu}$  are the cells of volume  $\mu$  in the geometric ordering.

We store this ordering by storing a pointer from each cell  $C_i = (\ell, x_1, \dots, x_d)$  to its successor  $C_{i+1} = (\ell, x'_1, \dots, x'_d)$ , which allows to scan the cells  $C_1, \dots, C_{1/\mu}$  in linear time. For any point  $x \in P$ , using the floor function we can determine in time  $O(1)$  the cell  $(\ell, x_1, \dots, x_d)$  of volume  $\mu$  that  $x$  belongs to (in our machine model we assume that the floor function can be computed in constant time). This allows to determine the numbers  $|C_i \cap P|$  for all  $i$  in time  $O(|P| + 1/\nu)$ . We also compute each prefix sum  $s_i := \sum_{j < i} |C_j \cap P|$  and store it at cell  $C_i = (\ell, x_1, \dots, x_d)$ . Using an array  $A[\cdot]$  of size  $|P|$ , we store (a pointer to) the  $k$ -th point in  $C_i \cap P$  at position  $A[s_i + k]$ . This preprocessing can be performed in time  $O(|P| + 1/\nu)$ .

A given cell  $C$  of volume at least  $\nu$  may consist of several cells of volume  $\mu$ . By Lemma 8, these cells form a contiguous subsequence  $C_i, C_{i+1}, \dots, C_{j-1}, C_j$  of  $C_1, \dots, C_{1/\mu}$ , so that the points  $C \cap P$  form a contiguous subsequence of  $A$ . For constant access time, we store for each cell  $C$  of volume at least  $\nu$  the indices  $s_C, e_C$  of the first and last point of  $C \cap P$  in  $A$ . Then  $|C \cap P| = e_C - s_C + 1$  and the  $k$ -th point in  $C \cap P$  is stored at  $A[s_C + k]$ . Thus, both queries can be answered in constant time. Note that the ordering  $A[\cdot]$  of the points in  $C \cap P$  is a mix of the geometric ordering of cells of volume  $\mu$  and the given ordering of  $P$  within a cell of volume  $\mu$ , in particular this ordering indeed only depends on  $P$  and  $\nu$ . ◀

Next we construct a partitioning of  $\mathbb{T}^d \times \mathbb{T}^d$  into products of cells  $A_i \times B_i$ . This partitioning allows to split the problem of sampling the edges of a GIRG into one problem for each  $A_i \times B_i$ , which is beneficial, since each product  $A_i \times B_i$  has one of two easy types. For any  $A, B \subseteq \mathbb{T}^d$  we denote the distance of  $A$  and  $B$  by  $d(A, B) = \inf_{a \in A, b \in B} \|a - b\|$ .

► **Lemma 10.** *Let  $0 < \nu \leq 1$ . In time  $O(1/\nu)$  we can construct a set*

$\mathcal{P}_\nu = \{(A_1, B_1), \dots, (A_s, B_s)\}$  such that

- (1)  $A_i, B_i$  are cells with  $\text{VOL}(A_i) = \text{VOL}(B_i) \geq \nu$ ,
- (2) for all  $i$ , either  $d(A_i, B_i) = 0$  and  $\text{VOL}(A_i) = \lceil \nu \rceil_{2^d}$  (type I) or  $d(A_i, B_i) \geq \text{VOL}(A_i)^{1/d}$  (type II),
- (3) the sets  $A_i \times B_i$  partition  $\mathbb{T}^d \times \mathbb{T}^d$ ,
- (4)  $s = O(1/\nu)$ .

**Proof.** Note that for cells  $A, B$  of equal volume we have  $d(A, B) = 0$  if and only if either  $A = B$  or (the boundaries of)  $A$  and  $B$  touch. For a cell  $C$  of level  $\ell$  we let  $\text{par}(C)$  be its *parent*, i.e., the unique cell of level  $\ell - 1$  that  $C$  is contained in. Let  $\mu = \lceil \nu \rceil_{2^d}$ . We define  $\mathcal{P}_\nu$  as follows. For any pair of cells  $(A, B)$  with  $\text{VOL}(A) = \text{VOL}(B) \geq \nu$ , we add  $(A, B)$  to  $\mathcal{P}_\nu$  if either (i)  $\text{VOL}(A) = \text{VOL}(B) = \mu$  and  $d(A, B) = 0$ , or (ii)  $d(A, B) > 0$  and  $d(\text{par}(A), \text{par}(B)) = 0$ .

Property (1) follows by definition. Regarding property (2), the pairs  $(A, B)$  added in case (i) are clearly of type I. Observe that two cells  $A, B$  of equal volume that are not equal or touching have distance at least the sidelength of  $A$ , which is  $\text{VOL}(A)^{1/d}$ . Thus, in case (ii) the lower bound  $d(A, B) > 0$  implies  $d(A, B) \geq \text{VOL}(A)^{1/d}$ , so that  $(A, B)$  is of type II.

For property (3), consider  $(x, y) \in \mathbb{T}^d \times \mathbb{T}^d$  and let  $A, B$  be the cells of volume  $\mu$  containing  $x, y$ . Let  $A^{(0)} := A$  and  $A^{(i)} := \text{par}(A^{(i-1)})$  for any  $i \geq 1$ , until  $A^{(k)} = \mathbb{T}^d$ . Similarly, define  $B = B^{(0)} \subset \dots \subset B^{(k)} = \mathbb{T}^d$  and note that  $\text{VOL}(A^{(i)}) = \text{VOL}(B^{(i)})$ . Observe that each set  $A^{(i)} \times B^{(i)}$  contains  $(x, y)$ . Moreover, any set  $A' \times B'$ , where  $A', B'$  are cells with  $\text{VOL}(A') = \text{VOL}(B')$  and  $(x, y) \in A' \times B'$ , is of the form  $A^{(i)} \times B^{(i)}$ . Thus, to show that  $\mathcal{P}_\nu$  partitions  $\mathbb{T}^d \times \mathbb{T}^d$  we need to show that it contains exactly one of the pairs  $(A^{(i)}, B^{(i)})$  (for any  $x, y$ ). To show this, we use the monotonicity  $d(A^{(i)}, B^{(i)}) \geq d(A^{(i+1)}, B^{(i+1)})$  and consider two cases. If  $d(A, B) = 0$  then we add  $(A, B)$  to  $\mathcal{P}_\nu$  in case (i), and we add no further  $(A^{(i)}, B^{(i)})$ , since  $d(A^{(i)}, B^{(i)}) = 0$  for all  $i$ . If  $d(A, B) > 0$  then since  $d(A^{(k)}, B^{(k)}) = d(\mathbb{T}^d, \mathbb{T}^d) = 0$  there is a unique index  $0 \leq i < k$  with  $d(A^{(i)}, B^{(i)}) > 0$  and  $d(A^{(i+1)}, B^{(i+1)}) = 0$ . Then we add  $(A^{(i)}, B^{(i)})$  in case (ii) and no further  $(A^{(j)}, B^{(j)})$ . This proves property (3).



---

**Algorithm 1** Sampling algorithm for GIRGs in expected time  $O(n)$ 


---

```

1:  $E := \emptyset$ 
2: sample the positions  $\mathbf{x}_v$ ,  $v \in V$ , and determine the weight layers  $V_i$ 
3: for all  $1 \leq i \leq L$  do build data structure  $\mathcal{D}_{\nu(i)}(\{\mathbf{x}_v \mid v \in V_i\})$  with  $\nu(i) := \frac{w_i w_0}{W}$ 
4: for all  $1 \leq i \leq j \leq L$  do
5:   construct partitioning  $\mathcal{P}_{\nu(i,j)}$  with  $\nu(i,j) := \frac{w_i w_j}{W}$ 
6:   for all  $(A, B) \in \mathcal{P}_{\nu(i,j)}$  of type I do
7:     for all  $u \in V_i^A$  and  $v \in V_j^B$  do with probability  $p_{uv}$  add edge  $\{u, v\}$  to  $E$ 
8:   for all  $(A, B) \in \mathcal{P}_{\nu(i,j)}$  of type II do
9:      $\bar{p} := \min \left\{ c \cdot \frac{1}{d(A,B)^{\alpha d}} \cdot \left( \frac{w_i w_j}{W} \right)^\alpha, 1 \right\}$ 
10:     $r := \text{Geo}(\bar{p})$ 
11:    while  $r \leq |V_i^A| \cdot |V_j^B|$  do
12:      determine the  $r$ -th pair  $(u, v)$  in  $V_i^A \times V_j^B$ 
13:      with probability  $p_{uv}/\bar{p}$  add edge  $\{u, v\}$  to  $E$ 
14:       $r := r + \text{Geo}(\bar{p})$ 
15:   if  $i = j$  then remove all edges with  $u > v$  sampled in this iteration

```

---

Property (4) follows from the running time bound of  $O(1/\nu)$ , which we show in the following. Note that we can enumerate all  $1/\mu = O(1/\nu)$  cells of volume  $\mu$ , and all of the at most  $3^d = O(1)$  touching cells of the same volume, in time  $O(1/\nu)$ , proving the running time bound for case (i). Moreover, we can enumerate all  $2^{\ell-d}$  cells  $C$  in level  $\ell$ , together with all of the at most  $3^d = O(1)$  touching cells  $C'$  in the same level. Then we can enumerate all  $2^d = O(1)$  cells  $A$  that have  $C$  as parent as well as all  $O(1)$  cells  $B$  that have  $C'$  as parent. This enumerates (a superset of) all possibilities of case (ii). Summing the running time  $O(2^{\ell-d})$  over all levels  $\ell$  with volume  $2^{-\ell-d} \geq \nu$  yields a total running time of  $O(1/\nu)$ . ◀

**Weight layers.** We set  $w_0 := w_{\min}$  and  $w_i := 2w_{i-1}$  for  $i \geq 1$ . This splits the vertex set  $V = [n]$  into *weight layers*  $V_i := \{v \in V \mid w_{i-1} \leq v < w_i\}$  for  $1 \leq i \leq L$  with  $L = O(\log n)$ . We write  $V_i^C$  for the restriction of weight layer  $V_i$  to cell  $C$ ,  $V_i^C := \{v \in V_i \mid \mathbf{x}_v \in C\}$ .

**Geometric random variates.** For  $0 < p \leq 1$  we write  $\text{Geo}(p)$  for a geometric random variable, taking value  $i \geq 1$  with probability  $p(1-p)^{i-1}$ .  $\text{Geo}(p)$  can be sampled in constant time using the simple formula  $\lceil \frac{\log(R)}{\log(1-p)} \rceil$ , where  $R$  is chosen uniformly at random in  $(0, 1)$ , see [30]. To evaluate this formula exactly in time  $O(1)$  we need to assume the RealRAM model of computation. However, also on a bounded precision machine like the WordRAM  $\text{Geo}(p)$  can be sampled in expected time  $O(1)$  [16].

### 3.2 The Algorithm

Given the model parameters, our Algorithm 1 samples the edge set  $E$  of a GIRG. To this end, we first sample all vertex positions  $\mathbf{x}_v$  uniformly at random in  $\mathbb{T}^d$ . Given weights  $w_1, \dots, w_n$  we can determine the weight layers  $V_i$  in linear time (we may use counting sort or bucket sort since there are only  $L = O(\log n)$  layers). Then we build the data structure from Lemma 9 for the points in  $V_i$  setting  $\nu = \nu(i) = \frac{w_i w_0}{W}$ , i.e., we build  $\mathcal{D}_{\nu(i)}(\{\mathbf{x}_v \mid v \in V_i\})$  for each  $i$ . In the following, for each pair of weight layers  $V_i, V_j$  we sample the edges between  $V_i$  and  $V_j$ . To this end, we construct the partitioning  $\mathcal{P}_{\nu(i,j)}$  from Lemma 10 with  $\nu(i,j) = \frac{w_i w_j}{W}$ . Since

$\mathcal{P}_{\nu(i,j)}$  partitions  $\mathbb{T}^d \times \mathbb{T}^d$ , every pair of vertices  $u \in V_i, v \in V_j$  satisfies  $\mathbf{x}_u \in A, \mathbf{x}_v \in B$  for exactly one  $(A, B) \in \mathcal{P}_{\nu(i,j)}$ . Thus, we can iterate over all  $(A, B) \in \mathcal{P}_{\nu(i,j)}$  and sample the edges between  $V_i^A$  and  $V_j^B$ .

If  $(A, B)$  is of type I, then we simply iterate over all vertices  $u \in V_i^A$  and  $v \in V_j^B$  and add the edge  $\{u, v\}$  with probability  $p_{uv}$ ; this is the trivial sampling algorithm. Note that we can efficiently enumerate  $V_i^A$  and  $V_j^B$  using the data structure  $\mathcal{D}_{\nu(i)}(\{\mathbf{x}_v \mid v \in V_i\})$  that we constructed above.

If  $(A, B)$  is of type II, then the distance  $\|x - y\|$  of any two points  $x \in A, y \in B$  satisfies  $d(A, B) \leq \|x - y\| \leq d(A, B) + \text{VOL}(A)^{1/d} + \text{VOL}(B)^{1/d} \leq 3d(A, B)$ , by the definition of type II. Thus,  $\bar{p} = \min \left\{ c \cdot \frac{1}{d(A, B)^{\alpha d}} \cdot \left( \frac{w_i w_j}{W} \right)^\alpha, 1 \right\}$  is an upper bound on the edge probability  $p_{uv}$  for any  $u \in V_i^A, v \in V_j^B$ , and it is a good upper bound since  $d(A, B)$  is within a constant factor of  $\|\mathbf{x}_u - \mathbf{x}_v\|$  and  $w_i, w_j$  are within constant factors of  $w_u, w_v$ . Now we first sample the set of edges  $\bar{E}$  between  $V_i^A$  and  $V_j^B$  that we would obtain if all edge probabilities were equal to  $\bar{p}$ , i.e., any  $(u, v) \in V_i^A \times V_j^B$  is in  $\bar{E}$  independently with probability  $\bar{p}$ . From this set  $\bar{E}$ , we can then generate the set of edges with respect to the true edge probabilities  $p_{uv}$  by throwing a coin for each  $\{u, v\} \in \bar{E}$  and letting it survive with probability  $p_{uv}/\bar{p}$ . Then in total we choose a pair  $(u, v)$  as an edge in  $E$  with probability  $\bar{p} \cdot (p_{uv}/\bar{p}) = p_{uv}$ , proving that we sample from the correct distribution. Note that here we used  $p_{uv} \leq \bar{p}$ . It is left to show how to sample the ‘‘approximate’’ edge set  $\bar{E}$ . First note that the data structure  $\mathcal{D}_{\nu}(\{\mathbf{x}_v \mid v \in V_i\})$  defines an ordering on  $V_i^A$ , and we can determine the  $\ell$ -th element in this ordering in constant time, similarly for  $V_j^B$ . Using the lexicographic ordering, we obtain an ordering on  $V_i^A \times V_j^B$  for which we can again determine the  $\ell$ -th element in constant time. In this ordering, the first pair  $(u, v) \in V_i^A \times V_j^B$  that is in  $\bar{E}$  is geometrically distributed, according to  $\text{Geo}(\bar{p})$ . Since geometric random variates can be generated in constant time, we can efficiently generate  $\bar{E}$ , specifically in time  $O(1 + |\bar{E}|)$ .

Finally, the case  $i = j$  is special. With the algorithm described above, for any  $u, v \in V_i$  we sample whether they form an edge twice, once for  $\mathbf{x}_u \in A, \mathbf{x}_v \in B$  (for some  $(A, B) \in \mathcal{P}_{\nu(i,j)}$ ) and once for  $\mathbf{x}_v \in A', \mathbf{x}_u \in B'$  (for some  $(A', B') \in \mathcal{P}_{\nu(i,j)}$ ). To fix this issue, in the case  $i = j$  we only accept a sampled edge  $(u, v) \in V_i^A \times V_j^B$  if  $u < v$ ; then only one way of sampling edge  $\{u, v\}$  remains. This changes the expected running time only by a constant factor.

### 3.3 Analysis

Correctness of our algorithm follows immediately from the above explanations. In the following we show that Algorithm 1 runs in expected linear time. This is clear for lines 1-2. For line 3, since building the data structure from Lemma 9 takes time  $O(|P| + 1/\nu)$ , it takes total time  $\sum_{i=1}^L O(|V_i| + W/(w_i w_0))$ . Clearly, the first summand  $|V_i|$  sums up to  $n$ . Using  $w_0 = w_{\min} = \Omega(1)$ ,  $W = O(n)$ , and that  $w_i$  grows exponentially with  $i$ , implying  $\sum_i 1/w_i = O(1)$ , also the second summand sums up to  $O(n)$ . For line 5, all invocations in total take time  $O(\sum_{i,j} W/(w_i w_j))$ , which is  $O(n)$ , since again  $W = O(n)$  and  $\sum_i 1/w_i = O(1)$ . We claim that for any weight layers  $V_i, V_j$  the expected running time we spend on any  $(A, B) \in \mathcal{P}_{\nu(i,j)}$  is  $O(1 + \mathbb{E}[|E_{i,j}^{A,B}|])$ , where  $E_{i,j}^{A,B}$  is the set of edges in  $V_i^A \times V_j^B$ . Summing up the first summand  $O(1)$  over all  $(A, B) \in \mathcal{P}_{\nu(i,j)}$  sums up to  $1/\nu(i, j) = W/(w_i w_j)$ . As we have seen above, this sums up to  $O(n)$  over all  $i, j$ . Summing up the second summand  $O(\mathbb{E}[|E_{i,j}^{A,B}|])$  over all  $(A, B) \in \mathcal{P}_{\nu(i,j)}$  and weight layers  $V_i, V_j$  yields the total expected number of edges  $O(\mathbb{E}[|E|])$ , which is  $O(n)$ , since the average weight  $W/n = O(1)$  and thus the expected average degree is constant.

It is left to prove the claim that for any weight layers  $V_i, V_j$  the expected time spent on  $(A, B) \in \mathcal{P}_{\nu(i,j)}$  is  $O(1 + \mathbb{E}[|E_{i,j}^{A,B}|])$ . If  $(A, B)$  is of type I, then any pair of vertices  $(u, v) \in$

$V_i^A \times V_j^B$  has probability  $\Theta(1)$  to form an edge: Since the volume of  $A$  and  $B$  is  $w_i w_j / W$ , their diameter is  $(w_i w_j / W)^{1/d}$  and we obtain  $\|x_u - x_v\| \leq (w_i w_j / W)^{1/d} = O((w_u w_v / W)^{1/d})$ , which yields  $p_{uv} = \Theta(\min\{(\frac{w_u w_v}{\|x_u - x_v\|^d W})^\alpha, 1\}) = \Theta(1)$ . As we spend time  $O(1)$  for any  $(u, v) \in V_i^A \times V_j^B$ , we stay in the desired running time bound  $O(\mathbb{E}[|E_{i,j}^{A,B}|])$ .

If  $(A, B)$  is of type II, we first sample edges  $\bar{E}$  with respect to the larger edge probability  $\bar{p}$ , and then for each edge  $e \in \bar{E}$  sample whether it belongs to  $E$ . This takes total time  $O(1 + |\bar{E}|)$ . Note that any edge  $e \in \bar{E}$  has constant probability  $p_{uv}/\bar{p} = \Theta(1)$  to survive: It follows from  $w_u = \Theta(w_i)$ ,  $w_v = \Theta(w_j)$ , and  $\|x_u - x_v\| = \Theta(d(A, B))$  that  $p_{uv} = \Theta(\bar{p})$ . Hence, we obtain  $\mathbb{E}[|\bar{E}|] = O(\mathbb{E}[|E_{i,j}^{A,B}|])$ , and the running time  $O(1 + |\bar{E}|)$  is in expectation bounded by  $O(1 + \mathbb{E}[|E_{i,j}^{A,B}|])$ . This finishes the proof of the claim.

#### 4 Stability of the Giant, Entropy, and Compression Algorithm

In this section we prove Theorems 5 and 6. More precisely, we show that w.h.p. the graph (and its giant) has separators of sublinear size, and we make use of these small separators to devise a compression algorithm that can store the graph using a linear number of bits in expectation. Note that the compression maintains only the graph up to isomorphism, not the underlying geometry. The main idea is to enumerate the vertices in an ordering that reflects the geometry, and then storing for each vertex  $i$  the differences  $i - j$  for all neighbors  $j$  of  $i$ . We start with a technical lemma that gives the number of edges intersecting an axis-parallel, regular grid. (For  $\gamma > 0$  with  $1/\gamma \in \mathbb{N}$ , the axis-parallel, regular grid with side length  $\gamma$  is the union of all  $d - 1$ -dimensional hyperplanes that are orthogonal to an axis and that are in distance  $k\gamma$  from the origin for a  $k \in \mathbb{Z}$ .) Both the existence of small separators and the efficiency of the compression algorithm follow easily from that formula. For detailed proofs we refer to the full version [18].

► **Lemma 11.** *Let  $\eta > 0$ . Let  $1 \leq \mu \leq n^{1/d}$  be an integer, and consider an axis-parallel, regular grid with side length  $1/\mu$  on  $\mathbb{T}^d$ . Then in expectation the grid intersects at most  $O(n \cdot (n/\mu^d)^{2-\beta+\eta} + (n^{2-\alpha} \mu^{d(\alpha-1)} + n^{1-1/d} \mu)(1 + \log(n/\mu^d)))$  edges.*

**Proof Outline.** For  $u, v \in V$ , let  $\rho_{uv}$  be the probability that the edge  $uv$  exists and cuts the grid. Let  $r_{\max} := 1/2$  be the diameter of  $\mathbb{T}^d$ . We write

$$\rho_{uv} = \int_0^{r_{\max}} \Pr[\|x_u - x_v\| = r] \cdot p_{uv}(r) \cdot \Pr[x_u, x_v \text{ in different cells of } \mu\text{-grid}] dr. \quad (1)$$

Observe that  $u$  and  $v$  have distance  $r$  with probability density  $\Pr[\|x_u - x_v\| = r] = O(r^{d-1})$ . For  $p_{uv}(r)$  we plug in the bound from (EP1) or (EP2), respectively. Finally, using symmetry of  $\mathbb{T}^d$  we can show that the last probability in (1) is bounded by  $O(\min\{\mu r, 1\})$ . The remainder of the proof is a straight-forward, yet technical calculation of the integral. ◀

**Compression algorithm.** We remark that our Theorem 6 does not directly follow from the general compression scheme on graphs with small separators in [5], since our graphs only have small separators in expectation, in particular, small subgraphs of size  $O(\sqrt{\log n})$  can form expanders and thus not have small separators. However, our algorithm loosely follows their algorithm as well as the practical compression scheme of [12], see also [22].

We first enumerate the vertices as follows. Recall the definition of cells from Section 2.4, and consider all cells of level  $\ell_0 := \lfloor \log n/d \rfloor$ . Note that the boundaries of these cells induce a grid as in Lemma 11. Since each such cell has volume  $\Theta(1/n)$ , the expected number of vertices in each cell is constant. We fix a geometric ordering of these cells as in Lemma 8,

---

**Algorithm 2** Finding the  $s$ -th neighbor of vertex  $i$ 


---

- |   |   |
|---|---|
| 1: $b := \text{SELECT}(i, B_V)$           | ▷ starting position of vertex $i$                         |
| 2: $k := \text{RANK}(b, B_E)$             | ▷ number of edges and vertices before $b$                 |
| 3: $b_1 := \text{SELECT}(k + s, B_E)$     | ▷ starting position of $s$ -th edge of vertex $i$         |
| 4: $b_2 := \text{SELECT}(k + s + 1, B_E)$ | ▷ bit after ending position of $s$ -th edge of vertex $i$ |
| 5: return $B[b_1 : b_2 - 1]$              | ▷ block that stores $s$ -th edge of vertex $i$            |
- 

and we enumerate the vertices in the order of the cells, breaking ties (between vertices in the same cell) arbitrarily. From now on we assume that the vertices are enumerated in this way, i.e., we identify  $V = [n]$ , where  $i \in [n]$  refers to the vertex with index  $i$ .

Having enumerated the vertices, for each vertex  $i \in [n]$  we store a block of  $1 + \deg(i)$  sub-blocks. The first sub-block consists of a single dummy bit (to avoid empty sequences arising from isolated vertices). In the other  $\deg(i)$  sub-blocks we store the differences  $i - j$  using  $\log_2 |i - j| + O(1)$  bits, where  $j$  runs through all neighbors of  $i$ . We assume that the information for all vertices is stored in a successive block  $B$  in the memory. Moreover, we create two more blocks  $B_V$  and  $B_E$  of the same length. Both  $B_V$  and  $B_E$  have a one-bit whenever the corresponding bit in  $B$  is the first bit of the block of a vertex, and  $B_E$  has also a one-bit whenever the corresponding bit in  $B$  is the first bit of an edge (i.e., the first bit encoding a difference  $i - j$ ). All other bits in  $B_V$  and  $B_E$  are zero.

It is clear that with the data above the graph is determined. To handle queries efficiently, we replace  $B_V$  and  $B_E$  each with a rank/select data structure. This data structure allows to handle in constant time queries of the form “Rank( $b$ )”, which returns the number of one-bits up to position  $b$ , and “Select( $i$ )”, which returns the position of the  $i$ -th one-bit [36, 27, 43]. Given  $i, s \in \mathbb{N}$ , we can find the index of the  $s$ -th neighbor of  $i$  in constant time by Algorithm 2. Note that we can also compute  $\deg(i)$  in constant time as  $\text{RANK}(b_{i+1}, B_E) - \text{RANK}(b_i, B_E) - 1$ , where  $b_i = \text{SELECT}(i, B_V)$  and  $b_{i+1} = \text{SELECT}(i + 1, B_V)$  are the starting positions of vertex  $i$  and  $i + 1$ , respectively. In particular, it is possible for Algorithm 2 to first check whether  $s \leq \deg(i)$ .

We need to show that the data structure needs  $O(n)$  bits in expectation. There are  $n$  dummy bits, so we must show that we require  $O(n)$  bits to store all differences  $i - j$ , where  $ij$  runs through all edges of the graph. We need  $\log_2 |i - j| + O(1)$  bits for each edge, and the  $O(1)$  terms sum up to  $O(|E|)$ , which is  $O(n)$  in expectation. Thus, it remains to prove the following.

► **Lemma 12.** *If  $V$  is enumerated geometrically, then  $\mathbb{E}[\sum_{ij \in E} \log(|i - j|)] = O(n)$ .*

**Proof outline.** The geometric ordering puts all the vertices that are in the same cell of a  $2^{-\ell}$ -grid in a consecutive block, for all  $1 \leq \ell \leq \ell_0$ . Therefore, if  $e = ij$  does not intersect the  $2^{-\ell}$ -grid then  $|i - j| \leq \#\{\text{vertices in the } 2^{-\ell}\text{-cell of } e\} \approx n2^{-\ell}$ . Hence, if  $E_{\ell+1}$  is the number of edges intersecting the  $2^{-\ell}$ -grid, but not the  $2^{-\ell-1}$ -grid, then  $\mathbb{E}[\sum_{ij \in E} \log(|i - j|)] \approx \sum_{\ell=0}^{\ell_0} \mathbb{E}[E_{\ell+1}] \log(n2^{-\ell})$ , and we show that the latter term is  $O(n)$  using Lemma 11. ◀

**Acknowledgements.** We thank Hafsteinn Einarsson, Tobias Friedrich, and Anton Krohmer for helpful discussions.

---

**References**


---

- 1 M. A. Abdullah, M. Bode, and N. Fountoulakis. Typical distances in a geometric model for complex networks. *Preprint available at arXiv:1506.07811*, 2015.

- 2 W. Aiello, A. Bonato, C. Cooper, J. Janssen, and P. Prałat. A spatial web graph model with local influence regions. *Internet Mathematics*, 5(1-2):175–196, 2008.
- 3 A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- 4 V. Batagelj and U. Brandes. Efficient generation of large random networks. *Physical Review E*, 71(3):036113, 2005.
- 5 D.K. Blandford, G.E. Blelloch, and I.A. Kash. Compact representations of separable graphs. In *Proceedings of the 14 annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 679–688. SIAM, 2003.
- 6 T. Bläsius, T. Friedrich, and A. Krohmer. Hyperbolic random graphs: Separators and treewidth. In *European Symposium on Algorithms (ESA)*, pages 15:1–15:16, 2016.
- 7 T. Bläsius, T. Friedrich, and A. Krohmer. Cliques in hyperbolic random graphs. *Algorithmica*, 2017.
- 8 T. Bläsius, T. Friedrich, A. Krohmer, and S. Laue. Efficient embedding of scale-free graphs in the hyperbolic plane. In *European Symposium on Algorithms (ESA)*, pages 16:1–16:18, 2016.
- 9 M. Bode, N. Fountoulakis, and T. Müller. On the giant component of random hyperbolic graphs. In *7th European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB)*, pages 425–429. Springer, 2013.
- 10 M. Bode, N. Fountoulakis, and T. Müller. On the geometrisation of the Chung-Lu model and its component structure. *Preprint*, 2014.
- 11 M. Boguñá, F. Papadopoulos, and D. Krioukov. Sustaining the Internet with hyperbolic mapping. *Nature Communications*, 1(6), September 2010.
- 12 P. Boldi and S. Vigna. The WebGraph framework I : compression techniques. In *13th International Conference on World Wide Web (WWW)*, pages 595–602, 2004.
- 13 B. Bollobás, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structures & Algorithms*, 31(1):3–122, 2007.
- 14 A. Bonato, J. Janssen, and P. Prałat. A geometric model for on-line social networks. In *1st International Workshop on Modeling Social Media (WOSM)*, 2010.
- 15 M. Bradonjić, A. Hagberg, and A.G. Percus. The structure of geographical threshold graphs. *Internet Mathematics*, 5(1-2):113–139, 2008.
- 16 K. Bringmann and T. Friedrich. Exact and efficient generation of geometric random variates and random graphs. In *40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 267–278, 2013.
- 17 K. Bringmann, R. Keusch, and J. Lengler. Average distance in a general class of scale-free networks with underlying geometry. *Preprint available at arxiv:1602.05712*, 2016.
- 18 K. Bringmann, R. Keusch, and J. Lengler. Sampling geometric inhomogeneous random graphs in linear time. *Preprint available at arXiv:1511.00576*, 2016.
- 19 K. Bringmann, R. Keusch, J. Lengler, Y. Maus, and A. Molla. Greedy routing and the algorithmic small-world phenomenon. *Preprint available at arXiv:1612.05539*, 2017.
- 20 E. Candellero and N. Fountoulakis. Clustering and the hyperbolic geometry of complex networks. In *11th International Workshop on Algorithms and Models for the Web Graph (WAW)*, pages 1–12, 2014.
- 21 E. Candellero and N. Fountoulakis. Bootstrap percolation and the geometry of complex networks. *Stochastic Processes and their Applications*, 126:234–264, 2016.
- 22 F. Chierichetti, R. Kumar, S. Lattanzi, M. Mitzenmacher, A. Panconesi, and P. Raghavan. On compressing social networks. In *15th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 219–228, 2009.

- 23 F. Chierichetti, R. Kumar, S. Lattanzi, A. Panconesi, and P. Raghavan. Models for the Compressible Web. In *50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 331–340, 2009.
- 24 F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences (PNAS)*, 99(25):15879–15882, 2002.
- 25 F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002.
- 26 F. Chung and L. Lu. The average distance in a random graph with given expected degrees. *Internet Mathematics*, 1(1):91–113, 2004.
- 27 D.R. Clark and I. Munro. Efficient suffix trees on secondary storage. In *Proceedings of the 7 annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 383–391. SIAM, 1996.
- 28 M. Deijfen, R. van der Hofstad, and G. Hooghiemstra. Scale-free percolation. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 49(3):817–838, 2013.
- 29 P. Deprez, R. S. Hazra, and M. V. Wüthrich. Inhomogeneous long-range percolation for real-life network modeling. *Risks*, 3(1), 2015.
- 30 L. Devroye. *Nonuniform random variate generation*. Springer, New York, 1986.
- 31 S.N. Dorogovtsev and J.F.F. Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, 2002.
- 32 T. Friedrich and A. Krohmer. On the diameter of hyperbolic random graphs. In *42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, Lecture Notes in Computer Science, 2015.
- 33 L. Gugelmann, K. Panagiotou, and U. Peter. Random hyperbolic graphs: degree sequence and clustering. In *39th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 573–585, 2012.
- 34 M. Heydenreich, T. Hulshof, and J. Jorritsma. Structures in supercritical scale-free percolation. *Preprint available at arXiv:1604.08180*, 2016.
- 35 E. Jacob and P. Mörters. A spatial preferential attachment model with local clustering. In *Algorithms and Models for the Web Graph*, pages 14–25. Springer, 2013.
- 36 G. Jacobson. Space-efficient static trees and graphs. In *30th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 549–554, 1989.
- 37 M. Kiwi and D. Mitsche. A bound for the diameter of random hyperbolic graphs. In *2015 Proceedings of the Twelfth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 26–39. SIAM, 2015.
- 38 M. Kiwi and D. Mitsche. Spectral gap of random hyperbolic graphs and related parameters. *Preprint available at arXiv:1606.02240*, 2016.
- 39 C. Koch and J. Lengler. Bootstrap percolation on geometric inhomogeneous random graphs. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 147:1–147:15, 2016.
- 40 J. C. Miller and A. Hagberg. Efficient generation of networks with given expected degrees. In *8th International Conference on Algorithms and Models for the Web Graph (WAW)*, pages 115–126, 2011.
- 41 I. Norros and H. Reittu. On a conditionally Poissonian graph process. *Advances in Applied Probability*, 38(1):59–75, 2006.
- 42 F. Papadopoulos, D. Krioukov, M. Boguñá, and A. Vahdat. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In *INFOCOM 2010. 29th IEEE International Conference on Computer Communications*, pages 1–9, March 2010.
- 43 M. Pătraşcu. Succincter. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 305–313, 2008.
- 44 M. Penrose. *Random geometric graphs*, volume 5. Oxford University Press Oxford, 2003.

- 45 U. Peter. *Random Graph Models for Complex Systems*. PhD thesis, ETH Zurich, 2014.
- 46 R. van der Hofstad and J. Komjathy. Explosion and distances in scale-free percolation. *arXiv preprint arXiv:1706.02597*, 2017.
- 47 M. Von Looz, C.L. Staudt, H. Meyerhenke, and R. Prutkin. Fast generation of dynamic complex networks with underlying hyperbolic geometry. *Preprint available at arXiv:1501.03545*, 2015.