Laboratoire J.A. Dieudonné, Université Côte d'Azur

École Normale Supérieure Paris-Saclay, Université Paris-Saclay

INTERNSHIP REPORT

RANDOM HYPERBOLIC GRAPHS

Antoine Barrier — antoine.barrier@ens-paris-saclay.fr

under the supervision of Dieter Mitsche — dmitsche@unice.fr

M1 Hadamard – Mathematics and Applications – april to july 2018

Contents

Notations

In this report, we use the following notations:

Notation	Description
d	Euclidean distance
$d_{\mathbb{H}}$	hyperbolic distance
$d_{\mathcal{G}}$	graph distance, also known as the geodesic distance
$\mathbb{B}(u,r)$	the open ball of center u and radius r, for the Euclidean distance
$\mathbb{B}_{\mathbb{H}}(u,r)$	the open ball of center u and radius r, for the hyperbolic distance
A_n a.a.s.	$\mathbb{P}(A_n) \underset{n\to+\infty}{\longrightarrow} 1$ (a.a.s means *asymptotically almost surely*)
$X \lesssim Y$	Y stochastically dominates X

In this report we use *polar coordinates*. The natural use of polar coordinates emerges from the generation of random hyperbolic graphs (see Definition 2.1). For two points u, v, we denote by $\theta_{uv} \in [0,\pi]$ the (non oriented) rotation angle at the origin O between u and v.

We will also use the following LANDAU notations:

$$
\begin{aligned}
f = o(g) &\iff \forall k > 0, \exists n_0, \forall n \geq n_0, |f(n)| < k\,|g(n)| \\
f = \mathcal{O}(g) &\iff \exists k > 0, \exists n_0, \forall n \geq n_0, |f(n)| \leq k\,|g(n)| \\
f = \Theta(g) &\iff \exists k_1, k_2 > 0, \exists n_0, \forall n \geq n_0, k_1\,|g(n)| \leq |f(n)| \leq k_2\,|g(n)| \\
f = \omega(g) &\iff \forall k > 0, \exists n_0, \forall n \geq n_0, |f(n)| \geq k\,|g(n)| \\
f = \Omega(g) &\iff \exists k > 0, \forall n_0, \exists n \geq n_0, |f(n)| \geq k\,|g(n)|
\end{aligned}
$$

1 Resume

1.1 English version

Under the supervision of Dieter Mitsche, I worked on random hyperbolic graphs at the J.A. Dieudonné laboratory in Nice. This is a recent model of graphs (introduced in 2010) satisfying numerous properties of complex networks that we can find in real networks such as social media (Facebook friendships map, ...) or the Internet graph (consisting of all the routers and theirs communications, see Paragraph 2.2.4). The idea behind this model is to generate vertices in the hyperbolic plane and to connect them in a similar ways as in geometric graphs, but using a hyperbolic distance. It is this underlying hyperbolic geometry that allows to obtain naturally numerous properties of complex networks (power-law, small-world phenomenon, sparsity, clustering, ...).

As the topic is recent, there are still many properties to prove, despite a significant research activity on this model. I was interested in a model of rumor spreading in random hyperbolic graphs: the push & pull model (see Paragraph 2.2.6). My objective was to find the average time needed to inform the whole giant component of the graph (the component with the most vertices). I have been helped in this task by Dieter and one of his co-author, Marcos Kiwi, who is a researcher at the university of Chile and is interested by similar questions.

Part 2 of this report is a short introduction to random hyperbolic graphs, with a description the internship topic. Then Part 3 is divided into two parts: the first one presents simulations done during the internship, while the other one proposes a theorical approach of the rumor spreading model.

1.2 French version

Sous la direction de Dieter Mitsche, j'ai travaillé sur des graphes aléatoires hyperboliques au laboratoire J.A. Dieudonné de Nice. Il s'agit d'un modèle de graphes aléatoires récent (introduit en 2010) satisfaisant de nombreuses propriétés des graphes que l'on peut obtenir sur des réseaux sociaux (graphe des amitiés de Facebook, ...) ou même le graphe des routeurs/serveurs d'internet et leurs communications entre eux (voir Paragraphe 2.2.4). L'idée derrière ce modèle est de générer des points dans un plan hyperbolique puis de les connecter similairement aux graphes géométriques mais en utilisant une distance hyperbolique. C'est cette géométrie hyperbolique qui permet de faire ressortir naturellement de nombreuses propriétés des graphes dits complexes (graphes en loi de puissance, phénomène de petit monde, faible densité, fort coefficient de clustering).

Le sujet étant récent, il y a encore de très nombreuses propriétés à trouver et démontrer, malgré une forte activité de recherche sur ce modèle. Pour ma part, je me suis intéressé à un modèle de propagation d'une rumeur dans ces graphes : le modèle push & pull (voir Paragraphe 2.2.6). Mon objectif était de trouver le temps moyen nécessaire avant que tous les sommets de la composante géante du graphe (la composante avec le plus de sommets) ne soient informés. J'ai été aidé pour cela par mon maitre de stage ainsi que l'un de ses co-auteurs, Marcos Kiwi, qui est chercheur à l'université du Chili et s'intéresse à des questions similaires.

Dans ce rapport, vous trouverez une courte introduction sur les graphes aléatoires hyperboliques ainsi qu'une description du sujet (Partie 2). Puis la Partie 3 sera constituée d'une première section présentant les simulations que j'ai réalisées durant ce stage, et d'une seconde portant sur notre approche théorique du modèle de propagation de rumeur.

2 Context

2.1 Location and organisation of the intership

I did my internship in the J.A. DIEUDONNÉ laboratory of Nice. It is a university laboratory located in the Valrose Park in the north of the city. I joined the probability and statistics team (about 30 researchers and PhD students). More specifically, I was supervised by Dieter Mitsche. I also worked with Marcos Kiwi, one of Dieter's co-author, from the university of Chile. Marcos came to Nice for a week in July.

During the four months, I assisted many seminar talks, on the one hand talks about probabilistic or statistical research topics each week, and on the other hand thesis presentations of PhD students who were practicing their oral skills before the real defense.

2.2 Internship topic

During this internship, I studied random hyperbolic graphs. The objective was to find some properties about this new kind of complex random graphs introduced in [KPK+10] by Krioukov et al. in 2010.

In this section, I first introduce random hyperbolic graphs, then I recall some significant results and finally I present the problem I was interested in.

2.2.1 Complex networks

Complex networks refer to large networks satisfying the following fundamental properties :

1. they are *scale free*: their degree distribution follows a power law distribution for a big range of degrees, that is to say that for an important number of k, $\mathrm{card}(\{\text{nodes of degree } k\}) \sim Ck^{-\beta}$ for some constant C and an exponent β.
2. they have a high *clustering coefficient*: two nodes of the network that have a common neighbour are somewhat more likely to be connected with each other,
3. they are *sparse* (or *diluted*): the number of edges is proportional to the number of nodes,
4. they exhibit the *small world phenomenon*: almost all pairs of vertices that are in the same component are within a short distance from each other.

Complex networks have been studied a lot for about 20 years as most of large networks emerging in the real world (social media, internet, biology, ...) are complex networks. Their study has many applications, and finding models of graphs satisfying the four above points is a challenging problem. This is the case of random hyperbolic graphs introduced by Krioukov et al.

2.2.2 Random hyperbolic graphs

Random hyperbolic graphs are a generalisation of random geometric graphs in hyperbolic spaces. Indeed, random geometric graphs have nodes uniformly generated in some part of the Euclidean space and two vertices are connected by an edge if and only if the euclidian distance between both is less or equal to a fixed threshold which is a parameter of the model. Edges of random hyperbolic graphs are following the same rule, except that the distance used to generate edges is the hyperbolic distance $d_{\mathbb{H}}$ in a hyperbolic space of negative curvature.

Definition 2.1. [Random hyperbolic graph]
Let $\alpha, \nu > 0$ and $n \in \mathbb{N}$. Fixing $R := 2\log(n/\nu)$, we say that a graph $\mathcal{G}_{\alpha,\,\nu}(n) := (V, E)$ (with $|V| = n$) is a *random hyperbolic graph* if:

- the vertices $\{v_i\}_{1\leq i\leq n}$ of V are independently generated in $\mathbb{B}(O, R)$ with density[a]

$$f : \begin{array}{ccc} \mathbb{R}_+ \times [0, 2\pi[& \longrightarrow & \mathbb{R}_+ \\ (r, \theta) & \longmapsto & \frac{1}{2\pi}\rho(r) \end{array} \quad \text{where} \quad \rho(r) = \frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R) - 1}\mathbb{1}_{[0,R[}(r)$$

- the edge set E is entirely determined by the positions of the vertices, that is to say that two vertices $u, v \in V$ are adjacents if and only if $d_{\mathbb{H}}(u, v) \leq R$:

$$\{u, v\} \in E \iff d_{\mathbb{H}}(u, v) \leq R \tag{1}$$

[a] it means that r and θ are independent, r generated with density ρ and θ uniformly chosen in $[0, 2\pi[$

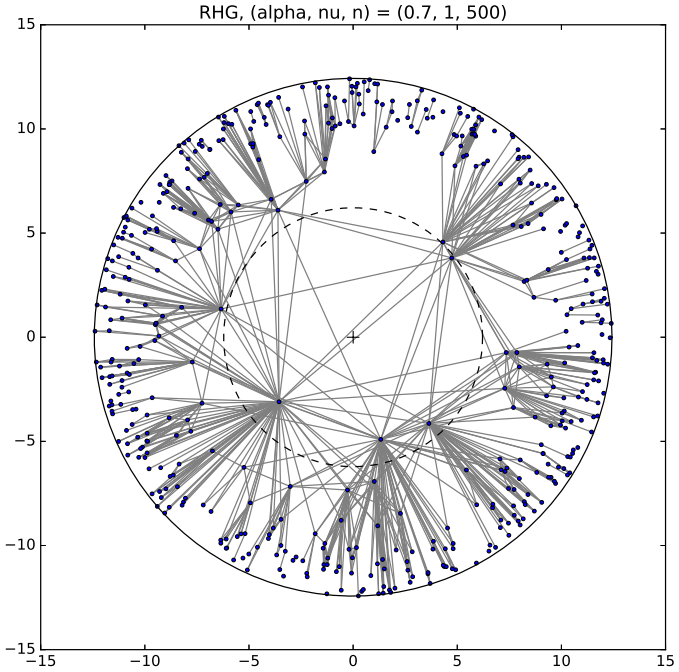


Figure 2.1: $\mathcal{G}_{0.7,\,1}(500)$. The solid (resp. segmented) circle is the boundary of $\mathbb{B}(O, R)$ (resp. $\mathbb{B}(O, R/2)$).

Remark 2.1.

- It is equivalent to say that the vertices are uniformly and independently generated in the ball $\mathbb{B}_{\mathbb{H}}(O, R)$ of the hyperbolic space of curvature $K = -\alpha^2$.
- The definition of the hyperbolic distance $d_{\mathbb{H}}$ can be found in a geometry course. This is the distance of the hyperbolic space of curvature $K = -1$. Nevertheless, we can use this **fundamental hyperbolic law of cosinus** that allows to check easily if condition (1) is satisfied or not: if $u = (r_u, \theta_u)$ and $v = (r_v, \theta_v)$, then

$$\boxed{\cosh d_{\mathbb{H}}(u, v) = \cosh r_u \cosh r_v - \sinh r_u \sinh r_v \cos\theta_{uv}} \tag{2}$$

- In their original paper, Krioukov et al. introduced a more general model with another parameter, called the temperature:it is a positive constant that controls the probability of connectivity: two nodes at hyperbolic distance d are connected by an edge with probability $1/(1+\mathrm{e}^{\frac{d-R}{2T}})$. We consider here the 0-temperature case which is the most studied case, as it is the easiest.

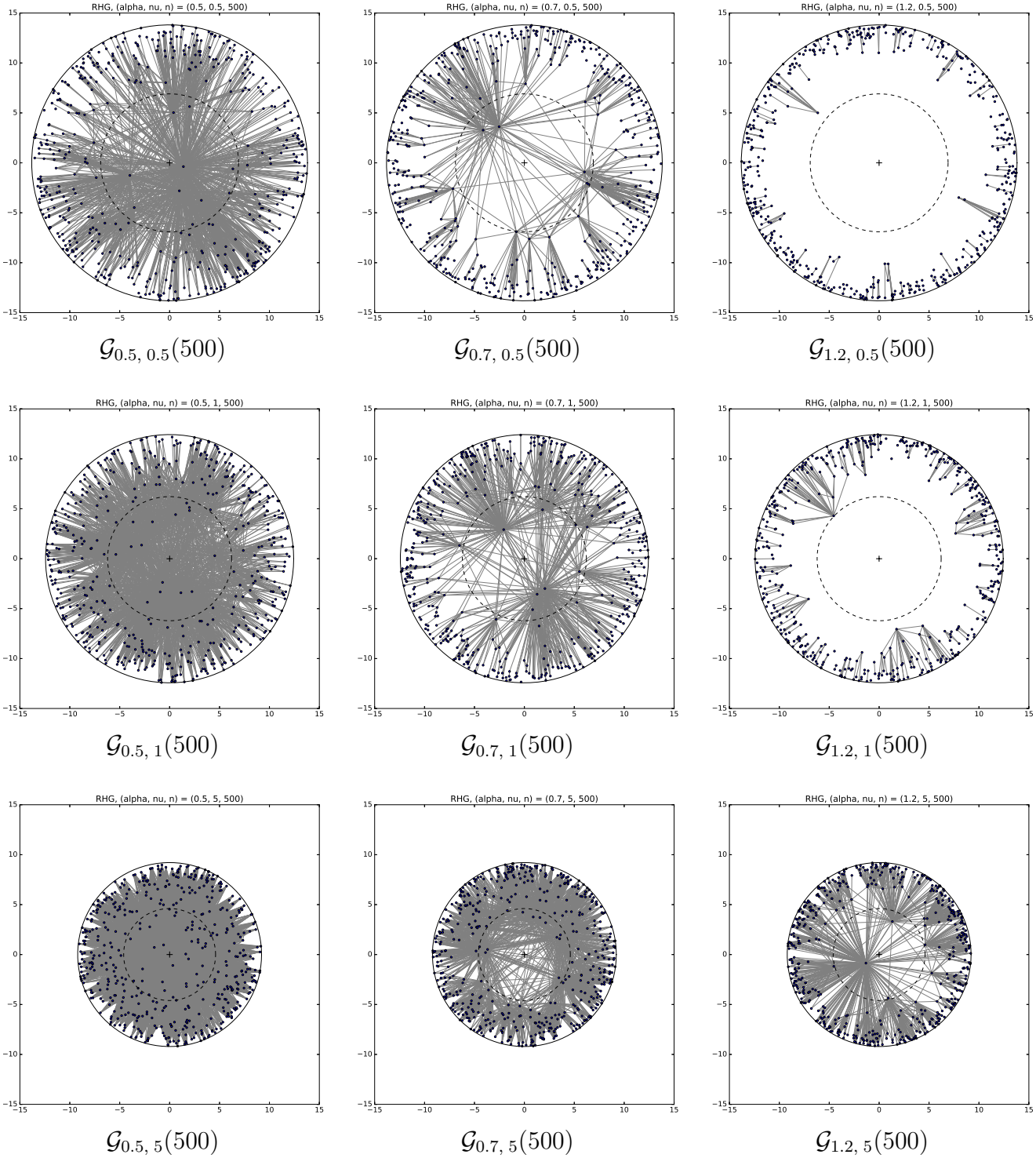


Figure 2.2: Some random hyperbolic graphs with different parameters α, ν. Note that the radii decrease in each new line, as ν increases.

2.2.3 Role of α and ν

In Figure 2.2, we can see examples of random hyperbolic graphs for different values of α and ν.
We observe the role of the two parameters :

- α controls the repartition of vertices near to the boundary : when α increases, vertices tend to accumulate at the boundary.
- ν controls the density of the graph, as the radius R depends on ν: when ν increases, the radius decreases and so the graph is more dense.

We will see below that α and ν control important properties of the model.

2.2.4 An example: the internet graph

In 2010, Boguñá, Papadopoulos and Krioukov showed that a famous complex network of our real world could be seen as a hyperbolic graph. They published a famous article in Nature Communications (see [BPK10]) concerning the Internet network: they considered the graph whose vertices are all the Internet routers (and also more generally all "Autonomous Systems" in the article), and with an edge between two connected routers.
They realised a maximum likelihood estimation to obtain the graph of Figure 2.3:

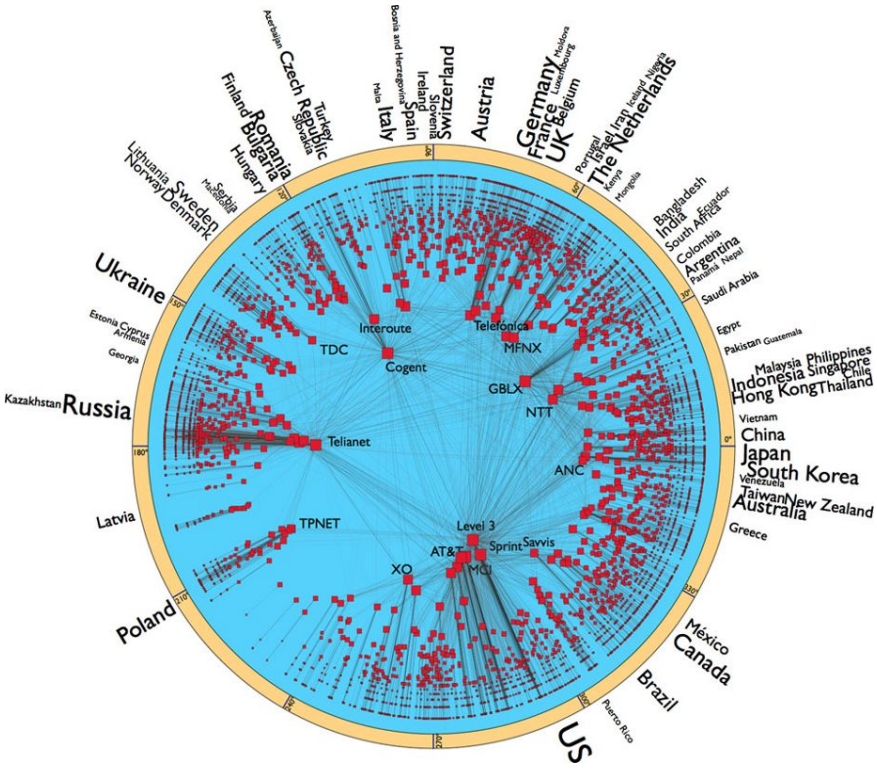


Figure 2.3: The hyperbolic map of the Internet. The size of nodes is proportional to the logarithm of their degrees. For the sake of clarity, only nodes with degree above 3 are shown.

There are 23752 vertices and 58416 links. Maximum likelihood estimation gives $\alpha = 0.55$, $R = 27$ and a temperature $T = 0.69$.
Nevertheless, we can raise doubts about the model, as many countries seem to be misplaced (see for instance the position of some European countries like Greece and Poland, or also India).

2.2.5 Known results

As random hyperbolic graphs have been introduced recently, there are many open questions about them. However, this is an active research field, many papers have been published in the last years with important results. First, it has been shown that random hyperbolic graphs satisfy the properties of complex networks introduced in Paragraph 2.2.1.

In the original paper of Krioukov et al., it has been shown that random hyperbolic graphs are scale-free (see point 1). They also found the exponent of the power-law distribution:

Theorem 2.1. [RHG are scale-free] *[KPK+10, Section IV]*
Let $\alpha, \nu > 0$. Then $\mathcal{G}_{\alpha,\nu}(n)$ admits a power-law degree distribution with exponent

$$\beta = \begin{cases} 2\alpha + 1 & \text{if } \alpha \geq \frac{1}{2} \\ 2 & \text{if } \alpha \leq \frac{1}{2} \end{cases}$$

We see that α controls the exponent of the power-law. This is not a surprise, as α controls ρ and it is harder to be connected to other points when you are close to the boundary of $\mathbb{B}(O, R)$. Note in particular that the exponent belongs to $[2, 3]$ when $\alpha \leq 1$ (we will see later that the case $\alpha > 1$ is less interesting). In [GPP12], Gugelmann et al. then showed that the expected degree distribution follows a power-law distribution *at all scales*, up to the maximum degree. By summing the degrees, one can show that the number of edges is $\mathcal{O}(n)$, which gives the sparsity of the model (point 3). In the same paper, they also showed that random hyperbolic graphs have a high clustering coefficient (point 2).

Then, Kiwi & Mitsche proved in [KM14] that the diameter of random hyperbolic graphs was at most poly-logarithmic, which in a sense means that random hyperbolic graphs exhibit a small-world phenomenon (point 4):

Theorem 2.2. [Diameter of random hyperbolic graphs] *[KM14, Theorems 12]*
Let $\alpha \in]1/2, 1[$ and $\nu > 0$. A.a.s., any two vertices u and v belonging to the same connected component satisfy

$$d_{\mathcal{G}}(u, v) = \mathcal{O}(\log^{C_0+1+o(1)}(n))$$

where $C_0 = \frac{4}{1-1.5\alpha+\alpha^2/2}$.

Friedrich and Krohmer improved the result in [FK15][1].

In addition, I have to mention some other results. The first one is about the size of the *giant component* (the largest connected component of the graph):

Theorem 2.3. [Size of the giant component] *[BFM15, Theorems 1.4 and 1.5]*
Let $\alpha, \nu > 0$. Denote by C_1 the giant component of $\mathcal{G}_{\alpha,\nu}(n)$, the following hold:

(i) if $\alpha > 1$, then $|C_1| < 8R^2 \log^3(R) n^{1/\alpha}$ a.a.s.,
(ii) if $\alpha < 1$ then there exists $c = c(\alpha, \nu)$ such that $|C_1| > cn$ a.a.s.,
(iii) if $\alpha = 1$, there exist constants $\frac{\pi}{8} \leq \nu_0 \leq \nu_1 \leq 20\pi$ such that the following hold:
- *if $\nu < \nu_0$, then $|C_1| \leq \frac{n}{\log(\log(R))}$ a.a.s.,*
- *if $\nu > \nu_1$, then $|C_1| \geq \frac{n}{610}$ a.a.s..*

When $\alpha < 1$, the size of the giant component is at least linear, which means a significant proportion of nodes are in the same component. When $\alpha > 1$, the size of the component is sub-linear, thus the study

[1] the paper of Friedrich and Krohmer however contained a mistake, and this was then corrected by Müller and Staps in [MS17]

of rumor spreading in the graph does not really make sense. There is a phase transition at $\alpha = 1$ at which the behaviour depends on ν.
In [KM17], Kiwi and Mitsche then showed that the second component is always sub-linear.

Another result is about the probability of connectivity:

Theorem 2.4. [probability of connectivity] *[BFM16, Theorem 1.2]*
Let $\alpha, \nu > 0$. Then the following hold :

(i) if $\alpha > 1/2$ then $\mathcal{G}_{\alpha,\nu}(n)$ is a.a.s. disconnected,
(ii) if $\alpha < 1/2$ then $\mathcal{G}_{\alpha,\nu}(n)$ is a.a.s. connected,
(iii) if $\alpha = 1/2$ then $\lim_{n\to+\infty} \mathbb{P}(\mathcal{G}_{\alpha,\nu}(n)$ is connected$) = f(\nu)$ where $f : \mathbb{R}^+ \longrightarrow]0,1]$ is a continuous function strictly increasing on $[0,\pi]$ and satisfying $f_{|[\pi,+\infty[} = 1$ and $f(0^+) = 0$.

When $\alpha < 1/2$, we have a unique component while it is not the case for $\alpha > 1/2$. At the phase transition, when $\alpha = 1/2$, the behaviour depends on ν.

2.2.6 My work

I started my internship reading a lot of articles in order to know better the random hyperbolic graph model and some techniques used in the demonstration of important results. Then I made some simulations. The goal was to understand how to generate random hyperbolic graphs with an important number of vertices and to verify some of the main properties I found in the articles I read.

After this necessary work, I studied a rumor spreading model: the push & pull model.

The push & pull model We consider a classic model of propagation in the giant component of the graph. Starting by randomly choosing a vertex which is the first informed at time $t = 0$, at each time two processes happen simultaneously and independently :

- a *push* : each informed vertex chooses one of its neighbours uniformly and transmits him the information if it did not already have it,
- a *pull* : each non-informed vertex chooses one of its neighbours uniformly. It receives the information if this neighbour already has it.

Our goal was to find the average time needed to inform the whole giant component of $\mathcal{G}_{\alpha,\nu}(n)$. As we want to propagate the rumor to a significant number of vertices, we consider the case $\alpha < 1$, owing to Theorem 2.3. In practice, we will work with $\alpha \in]1/2, 1[$ as when $\alpha \leq 1/2$, there are two many vertices near to the center of the graph and thus the spread seems to be too fast and hard to control.

My first work was to implement simulations of the process. The goal was to understand how the rumor propagates in the graph. I also worked on the theoretical problem, trying to use what I observed in the simulations. In Figure 2.4, we can see an instance of one push & pull simulation. We will explain in details what we see in Paragraph 3.1.4.

I worked and shared a lot with Dieter and Marcos on the mathematical approach and conducted simulations on my own at the same time.

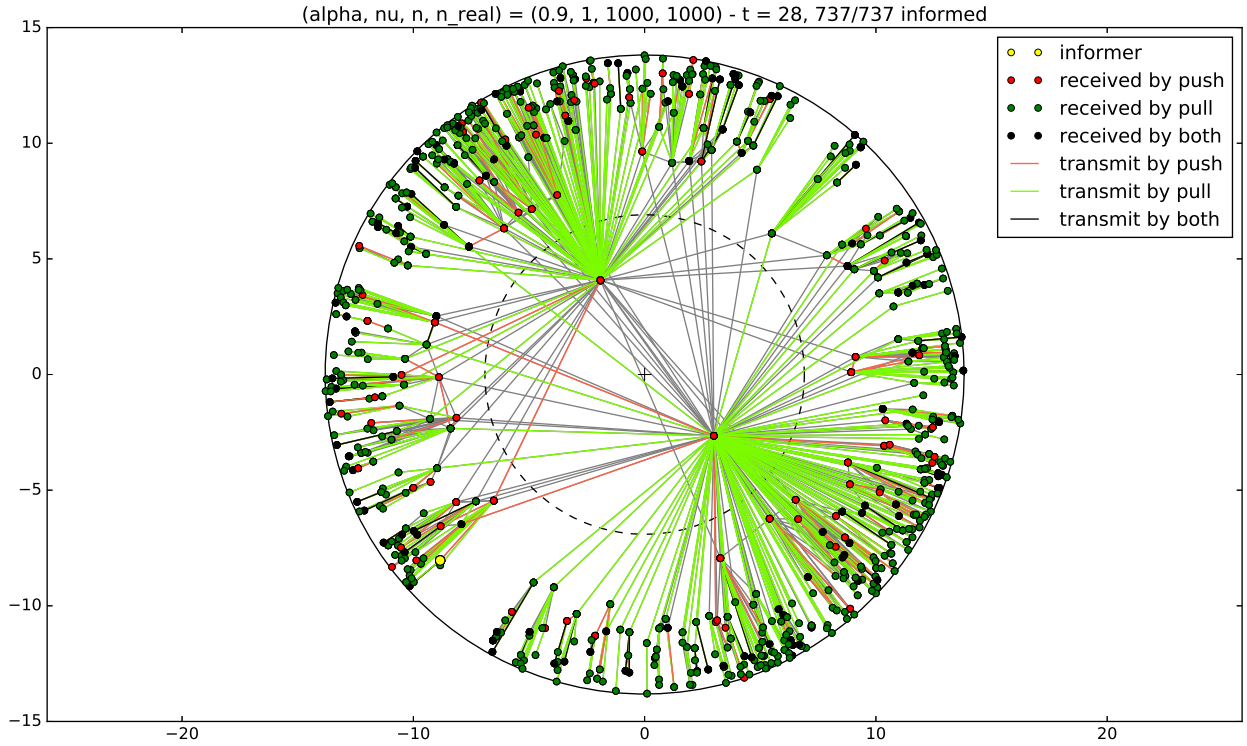


Figure 2.4: A push & pull rumor spreading in $\mathcal{G}_{0.9,1}(1000)$. Only vertices of the giant component are represented.

3 Report

This part contains what I did during this internship. It is composed of two sections: we begin with some simulations done with Python. Starting by the creation of a fast random hyperbolic graph generator, we then show some of the fundamental properties introduced in the last part. We also did simulations for the push & pull model, in order to better understand what happens in the graph. In the second section we present the theoretical work we started to work on. We will discuss about the push & pull rumor spreading in simplified models.

3.1 Simulations

After the reading of some articles, I started my internship by doing some simulations. It was necessary to understand better the model and its underlying hyperbolic geometry. Codes and simulation videos may be found at these links:

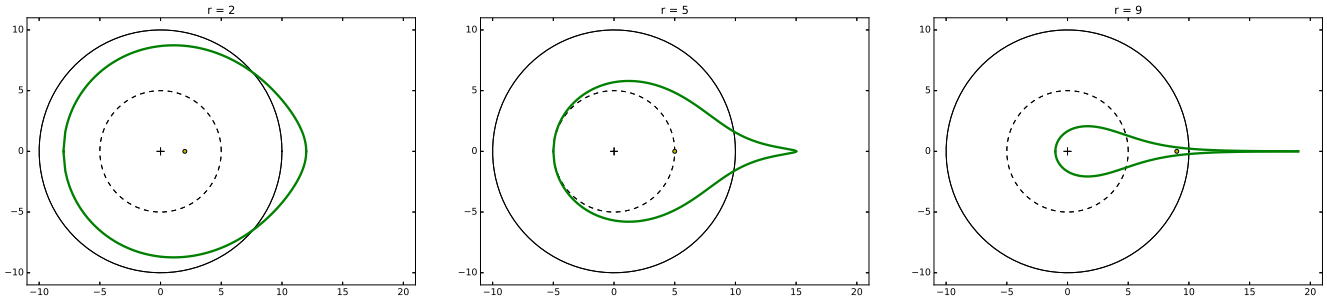
https://github.com/Tooine/RHG
https://www.youtube.com/playlist?list=PLVYgKw0bePxtkUTFZPzGhMyOiijz-BXkg

3.1.1 Balls

To begin with, let us have a look at the form of balls in the hyperbolic plane. Fixing $R > 0$, and $u = (r_u, \theta_u)$, we want to represent $\mathbb{B}_{\mathbb{H}}(u, R) = \{v \mid d_{\mathbb{H}}(u, v) \leq R\}$. We recall equation (2) :

$$\boxed{\cosh d_{\mathbb{H}}(u, v) = \cosh r_u \cosh r_v - \sinh r_u \sinh r_v \cos \theta_{uv}}$$

Note that $d_{\mathbb{H}}$ is rotation-invariant, so we just have to consider the case $\theta_u = 0$.
In the case where u is the origin O, this equation directly gives that $\mathbb{B}_{\mathbb{H}}(O, R) = \mathbb{B}(O, R)$. Otherwise we do not have this equality. Simulations give us the form of balls:



$r_u = 2$ $r_u = 5$ $r_u = 9$

Figure 3.1: $\mathbb{B}_{\mathbb{H}}(u, 10)$.

Recall that two vertices $u, v \in V$ are adjacent if and only if $d_{\mathbb{H}}(u, v) \leq R$. It is now easier to see which vertices are connected to a fixed vertex: a vertex with a small radial coordinate is almost connected to the whole space, while if it has a large radius, connected points will have a close angle or a really small radius (see Figure 3.2).

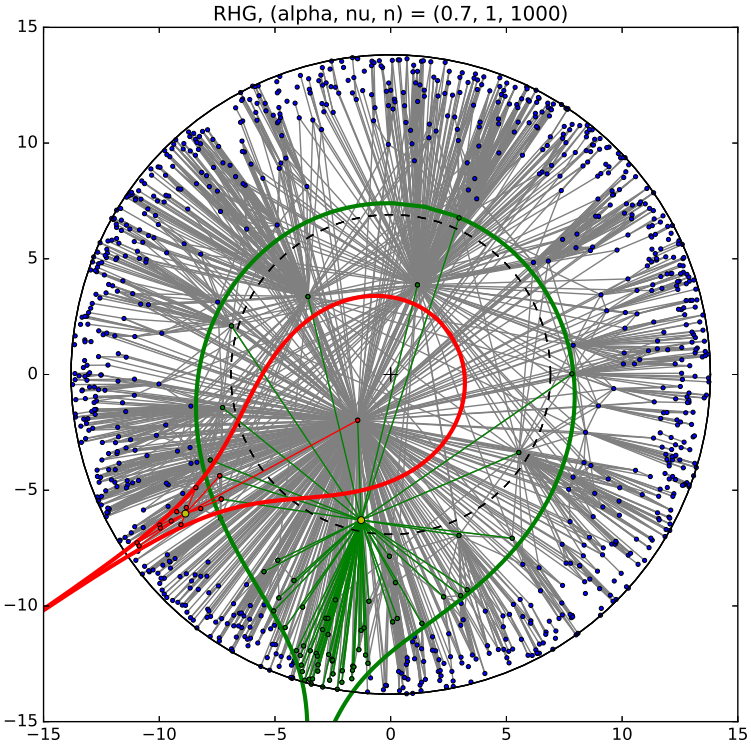


Figure 3.2: $\mathcal{G}_{0.7,\,1}(1000)$. For two vertices (in yellow), we highlight their connections and the hyperbolic ball of radius R around them. Observe that the vertex with green neighbours has more connections than the one with red neighbours, as its radius is smaller.

3.1.2 RHG generator

In order to check some properties with simulations – and also to find conjectures –, we need a quick random hyperbolic graph generator. As we have results for an asymptotical number of vertices, this generator has to be fast enough for big values of n.

Algorithm 1: greedy generator of $\mathcal{G}_{\alpha,\,\nu}(n)$

Input : α, ν, n
Output: V, E

$R \leftarrow 2\log(n/\nu)$
// V ordered list of vertices, E set of edges
$V \leftarrow []$
$E \leftarrow \{\}$
// Generation of V
for $i \in [\![1, n]\!]$**:**
  $r \leftarrow$ random radius with density $\rho = \rho_{\alpha,\nu}$
  $\theta \leftarrow$ random angle uniformly chosen in $[0, 2\pi[$
  V.addPoint((r, θ))
// Generation of E
for $i \in [\![1, n]\!]$**:**
  for $j \in [\![i+1, n]\!]$**:**
    if $d_{\mathbb{H}}(V[i], V[j]) \leq R$**:**
      E.addEdge($\{i, j\}$)

A naïve generation consists in generating vertices and then checking all pairwise hyperbolic distances between vertices and add an edge if it is smaller or equal to R (see Algorithm 1). The implementation is really simple, but its complexity is clearly in $\mathcal{O}(n^2)$, which is in practice bad, as it becomes hard to

generate a random hyperbolic graph with more than 1000 vertices.

To improve the complexity, we have to benefit from the geometry of the graph. An idea is to decompose the hyperbolic plane into circular bands. Most algorithms for generating random hyperbolic graphs are using this technique (see [Pen17, vÖLM16, BFKL16, BKL15, LS17]). I had two major problems with those algorithms: a large part is using parallel programmation, which I am not able to do in Python, and other algorithms are not well explained. Therefore I created my own generator using some ideas I found in those articles. It allows me to generate graphs with at most 10^6 vertices, while algorithms using parallelism can go up to 10^8 vertices.

Let us start by considering two radii r_1, r_2. We can see in Figure 3.2 that there exists an angle $\theta_{\max}$ such that two vertices u, v with $r_u = r_1$ and $r_v = r_2$ are adjacent if and only if $\theta_{uv} \in [0, \theta_{\max}]$.
Thus we need to introduce the function

$$\begin{array}{rccc} \Delta\theta: & [0,R]^2 & \longrightarrow & [0,\pi] \\ & r_1, r_2 & \longmapsto & \theta_{\max} = \max\{\theta \in [0,\pi] \,|\, d_{\mathbb{H}}((r_1,0),(r_2,\theta)) \leq R\} \end{array}$$

(the equality between $\theta_{\max}$ and the other term follows from the invariance by rotation of $d_{\mathbb{H}}$).
Given two vertices u and v, we have:

$$\{u,v\} \in E \quad \Longleftrightarrow \quad \theta_{uv} \leq \Delta\theta(r_u, r_v) \tag{3}$$

Using equation (2), we obtain:

$$\Delta\theta(r_1,r_2) = \arccos\left(\frac{\cosh(r_1)\cosh(r_2) - \cosh(R)}{\sinh(r_1)\sinh(r_2)}\right)$$

with the convention $\arccos([-\infty,-1]) = \pi$ and $\arccos([1,+\infty]) = 0$.
$\Delta\theta$ satisfies the following properties:

- $\Delta\theta$ is symmetric (in fact, if it was not, the graph would be directed ...),
- for all $r \geq 0$, $\Delta\theta(.,r)$ is decreasing (simply by derivating the term inside the $\arccos$),
- for all $r \geq 0$, $\Delta\theta(0,r) = \pi$, as the term inside the $\arccos$ is equal to $-\infty$. This is also logical, as a vertex at the origin is connected to all vertices.

We will use those properties to create a generator.

The idea is to decompose $\mathbb{B}(O,R)$ into circular bands. For any vertex, we can associate to each band an angle range in which all its neighbours located in that band are, as we can see in Figure 3.3. Then the objective is to use those ranges and a classification of vertices in each band to reduce the number of tests.
The main steps of the algorithm are the following. See Algorithm 2 for details:

lines 1-3 We decompose $\mathbb{B}(O,R)$ into circular bands. The first band is $\mathbb{B}(O,R/2)$ and then all bands have a constant length, as in Figure 3.3.

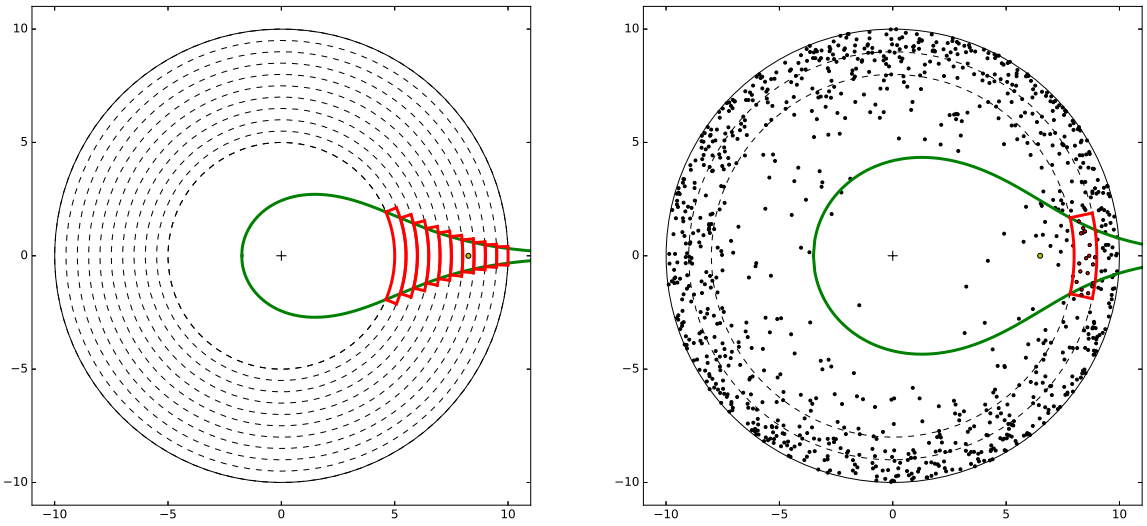
lines 4-17 Then we generate n vertices with density f[2]. Each vertex (r,θ) is added to its band and then we create requests for all upper bands (including its own band): for each band, we calculate the range of the request which is $[\theta \pm \Delta\theta(r, r_{\min})]$ where $r_{\min}$ is the inferior radius limit of the band[3]. All requests are added to a list of requests associated to each band.

Then we generate the set of edges:

lines 18-19 As vertices in $\mathbb{B}(O,R/2)$ are forming a clique (they are all connected, using the triangular inequality of $d_{\mathbb{H}}$), we add all couple of vertices in this ball to E.

[2] density function introduced in Definition 2.1

[3] Note that the range $[\theta \pm \Delta\theta(r, r_{\min})]$ is not always included in $[0, 2\pi]$. When it is not the case, and as we will then make comparisons with angles in $[0, 2\pi]$, we split the request in two requests with ranges included in $[0, 2\pi]$.



(a) For bands (length 0.5 here) of big radius, we can reduce the search of neighbours to the red area. All red areas are requests

(b) On each band (here band of length 1, radius 7 to 8), requests are sent to a small part of vertices (in the red area)

Figure 3.3: Circular bands decomposition and requests linked to a fixed vertex (in yellow).

line 21 For all bands, we sort the set of vertices (resp. requests) by their angle (resp. minimum angle).

lines 22-42 For each band (except the first one), we check the requests. Taking vertices (r, θ) in order (so sorted by angle), we create a list of all candidate requests such that θ is included in the range of the request. As requests are also sorted, we do not have to check this condition for all the requests. Then, for each candidate request, if the distance between the considered vertex and the vertex of the request is less than R, we add an edge between them, otherwise we do nothing[4]. For the next vertex, we delete candidate requests which are no more satisfying the conditions, and then add new satisfying requests.

In Figure 3.4, we can see the generation time of some random hyperbolic graphs with the two algorithms. For Algorithm 1, the number of comparisons does not depend on α and thus the time of execution does not depend either. For Algorithm 2, α has a little role. Indeed, when α decreases, there are more vertices near to the center of the graph. Those vertices create more requests (because they have more bands to test with) and moreover, theirs requests have bigger ranges.

By performing a linear regression, we see the order two ($\mathcal{O}(n^2)$) of Algorithm 1. Algorithm 2 seems to be of order one ($\mathcal{O}(n)$). In reality, it is not exactly the case. It is hard to find the complexity of the algorithm, as loops of the algorithm can depend on the random data (for instance the number of requests send by a vertex depends on its layer). Therefore we do not calculate a precise order, but we think it is almost linear with a logarithmic or polylogarithmic factor. See the references mentioned above for more details about the complexity of the best algorithms.

> REMARK 3.1. It takes a long time to plot the graph after the generation of V and E. But this is not a problem as most of the time, simulations do not require a plot of the graph.

[4] you can see in the algorithm other conditions (line 40) which are just here to avoid duplicates

Algorithm 2: fast generator of $\mathcal{G}_{\alpha,\,\nu}(n)$

Input : α, ν, n, β
Output: V, E

$R \leftarrow 2\log(n/\nu)$ $\quad V \leftarrow []$ $\quad E \leftarrow \{\}$
$nbBands \leftarrow \max(2, \lfloor \beta R \rfloor + 1)$
$limits \leftarrow [0, R/2, R/2 + c, R/2 + 2c, \ldots, R - c, R]$ with $c = R/[2 * (nbBands - 1)]$ *// Band Limits*
// Generation of V, band points and band requests
$bands\_V \leftarrow [[], [], \ldots, []]$ $\quad bands\_R \leftarrow [[], [], \ldots, []]$ *// two lists composed of $nbBands$ empty lists*
for $i \in [\![1, n]\!]$**:**
  $r \leftarrow$ random radius with density $\rho = \rho_{\alpha,\nu}$
  $\theta \leftarrow$ random angle uniformly chosen in $[0, 2\pi[$
  $b \leftarrow$ search band such that $limits[b] \leq r < limits[b+1]$
  V.addPoint((r, θ)) $\quad bands\_V$.addPoint((i, r, θ))
  // Generation of requests from point i
  for $band \in [\![\max(1, b), nbBands]\!]$**:**
    $\theta^{\min}_{req} \leftarrow \theta - \Delta\theta(r, limits[band])$ $\quad \theta^{\max}_{req} \leftarrow \theta + \Delta\theta(r, limits[band])$
    if $\theta^{\min}_{req} < 0$**:**
      $bands\_R[band]$.addRequests($[i, b, 0, \theta^{\max}_{req}], [i, b, \theta^{\min}_{req} + 2\pi, 2\pi]$) *// decomposition into 2 requests*
    elif $\theta^{\max}_{req} > 2\pi$**:**
      $bands\_R[band]$.addRequests($[i, b, 0, \theta^{\max}_{req} - 2\pi], [i, b, \theta^{\min}_{req}, 2\pi]$)
    else:
      $bands\_R[band]$.addRequest($[i, b, \theta^{\min}_{req}, \theta^{\max}_{req}]$)
// Generation of E: first we add the clique of vertices in band 0, then for other bands we analyse all requests
for *all couples* $(i, r_i, \theta_i), (j, r_j, \theta_j) \in bands\_V[0]$**:**
  E.addEdge($\{i, j\}$)
for $band \in [\![1, nbBands]\!]$**:**
  sort $bands\_V[band]$ by θ $\quad$ sort $bands\_R[band]$ by $\theta^{\min}_{req}$
  $candidates\_R = []$
  for $(i_{pt}, r_{pt}, \theta_{pt}) \in bands\_V[band]$**:**
    $new\_E = []$
    for $req \in candidates\_R$**:**
      if $\theta_{pt} > req[3]$**:**
        $candidates\_R$.removeRequest(req) *// remove requests whose range is exceeded*
    $test\_new\_req = True$
    while $test\_new\_req$ ***and*** $len(bands\_R[band]) > 0$**:**
      $req = (i_{req}, b_{req}, \theta^{\min}_{req}, \theta^{\max}_{req}) \leftarrow bands\_R[band][0]$
      if $\theta^{\min}_{req} \leq \theta_{pt}$**:**
        $bands\_R[band]$.removeRequest(req)
        if $\theta_{pt} \leq \theta^{\max}_{req}$**:**
          $candidates\_R$.addRequest(req) *// add new candidates requests for this vertex*
      else:
        $test\_new\_req = False$ *// other requests are also too high for this vertex*
    for $req = (i_{req}, b_{req}, \theta^{\min}_{req}, \theta^{\max}_{req}) \in candidates\_R$**:**
      $r_{req}, \theta_{req} = V[i_{req}]$
      $candidate = \{i_{req}, i_{pt}\}$
      if $d_{\mathbb{H}}((r_{req}, \theta_{req}), (r_{pt}, \theta_{pt})) \leq R$ ***and*** $candidate \notin new\_E$ ***and*** $i_{req} \neq i_{pt}$ ***and*** $((b_{req} = band$ ***and*** $i_{req} < i_{pt})$ ***or*** $b_{req} \neq band)$**:**
        new_E.addEdge($candidate$)
    E.addEdges(new_E)

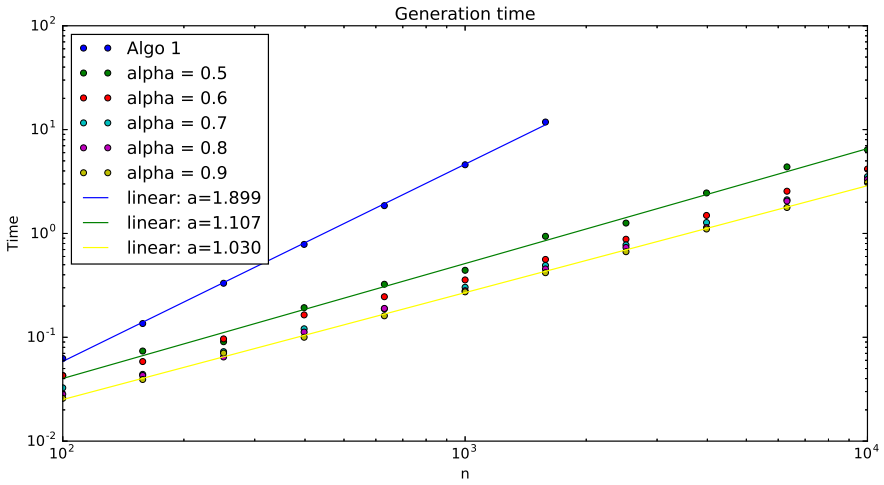


Figure 3.4: Generation time of $\mathcal{G}_{\alpha,1}(n)$.

3.1.3 Illustration of some properties

I also tried to verify some of the properties announced in Paragraph 2.2.5.

Theorem 2.1 We have seen that $\mathcal{G}_{\alpha,\,\nu}(n)$ admits a power-law distribution with exponent depending on α. Let us check this property by generating some random hyperbolic graphs and draw their degree distribution.

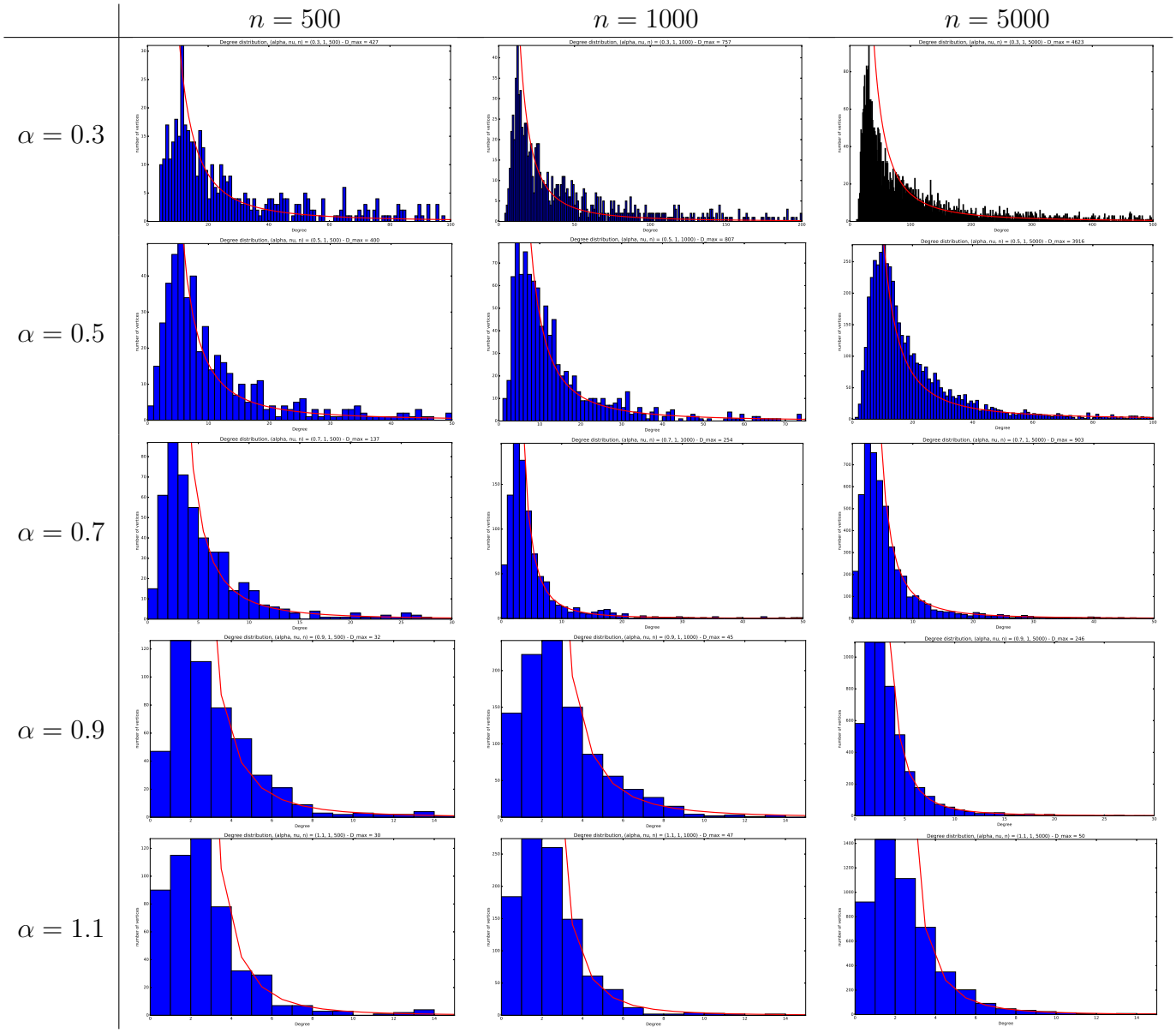


Figure 3.5: Degree distribution of vertices of $\mathcal{G}_{\alpha,\,1}(n)$. Theoretical distribution in red, up to a multiplicative contant. Distributions are cut to intervals of most interest.

In Figure 3.5, we can see the degree distribution of some random hyperbolic graphs. For each graph, we add the theoretical distribution in red, that is to say the curve $Ck^{-\beta}$ for a well chosen constant C and:

$$\beta = \begin{cases} 2\alpha + 1 & \text{if } \alpha \geq \frac{1}{2} \\ 2 & \text{if } \alpha \leq \frac{1}{2} \end{cases}$$

For the sake of clarity, we do not plot the whole distribution (so the maximum degree is not the limit of the horizontal axis).

Overall, we observe that the shape of the distribution corresponds to the expectation. For small values of α, there are small fluctuations but they do not modify the general shape.
Even if we do not plot the distribution entirely, in practice we observe that the match is up to the maximum degree, which confirms the result of Gugelmann et al. (see [GPP12]) who showed that the degree distribution follows a power-law distribution at all scales.

In Figure 3.6, we observe a concentration effect by considering the superposition of the degree distributions from 10 graphs: fluctuations are smaller.

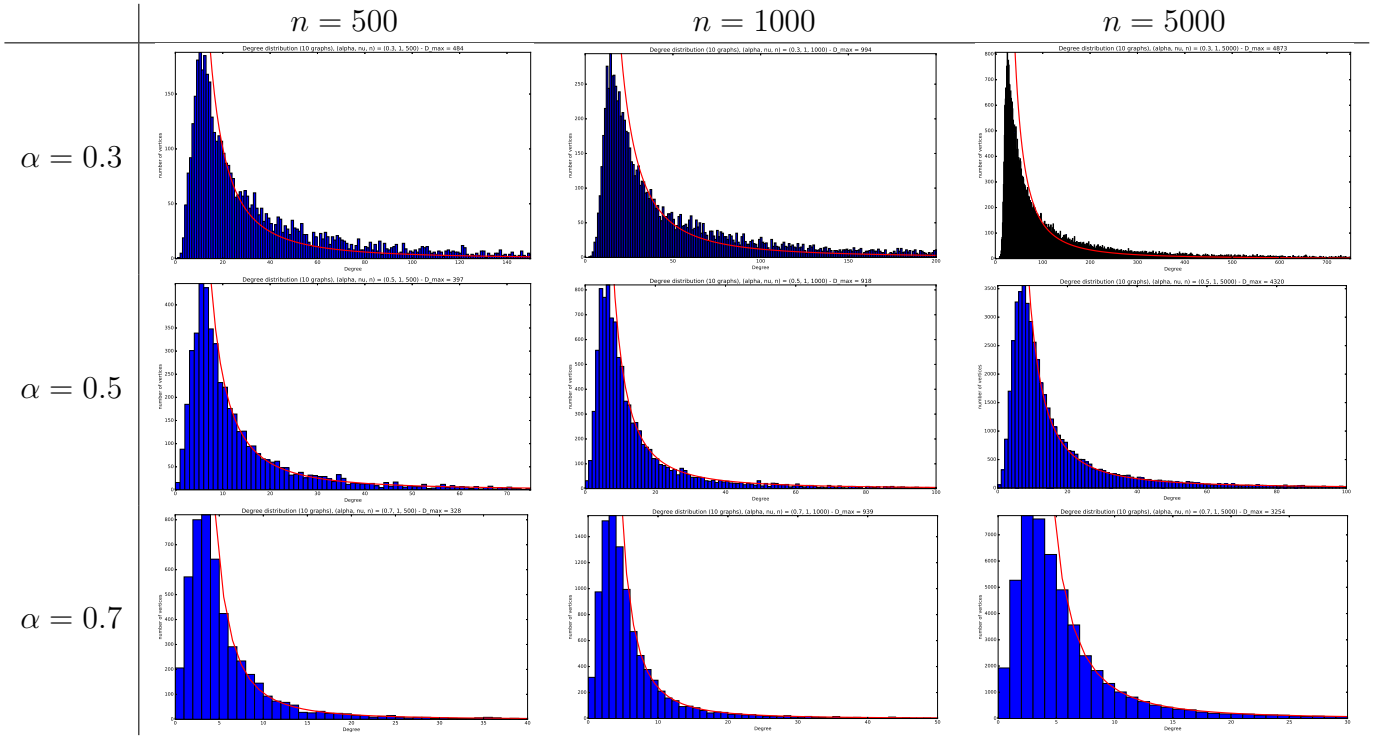


Figure 3.6: Concentration of the degree distribution: sum of 10 degree distributions of $\mathcal{G}_{\alpha,\,1}(n)$.

Taking other values of ν does not change the power-law distribution:

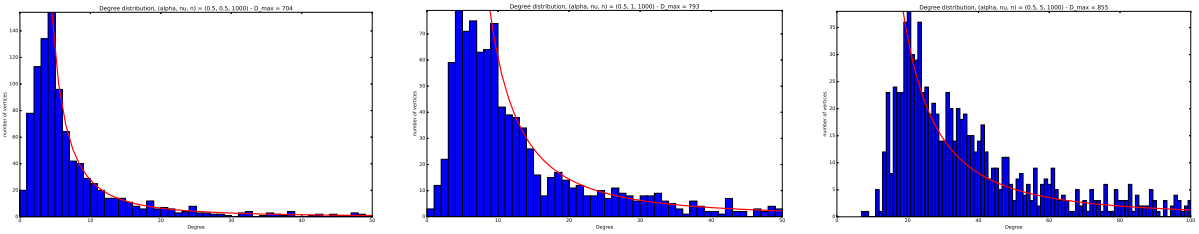
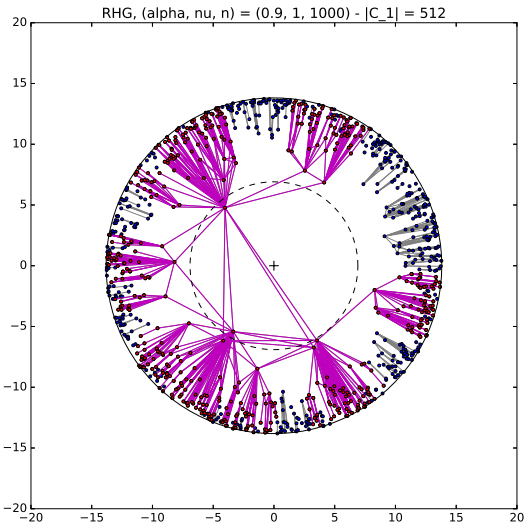


Figure 3.7: Degree distribution of $\mathcal{G}_{0.5,\,\nu}(1000)$.

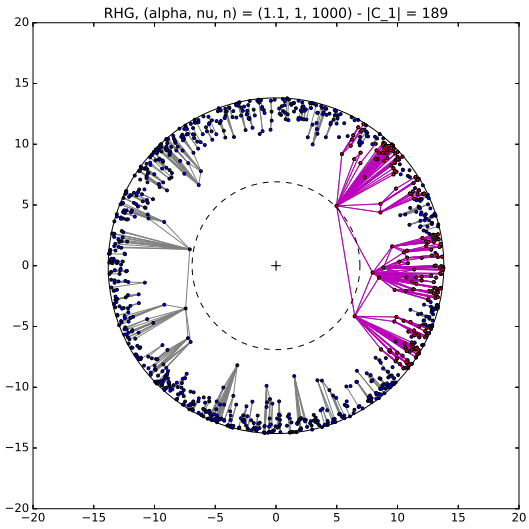
ν just modifies the constant C and the minimum degree where the power-law begins:

- if ν is big, then R is small and so the graph is more dense. Thus the number of vertices with a very small degree is really small, and the power-law distribution begins only after a minimum degree ($\simeq 20$ for $\nu = 5$),
- otherwise, if ν is small, the graph is more sparse and there are vertices with small degrees: the power-law distribution begins earlier ($\simeq 10$ for $\nu = 1$, $\simeq 4$ for $\nu = 0.5$).

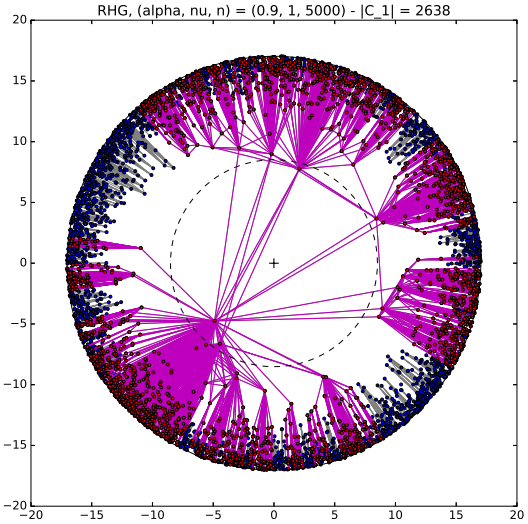
Theorem 2.3 We can also verify the growth properties about the giant component.



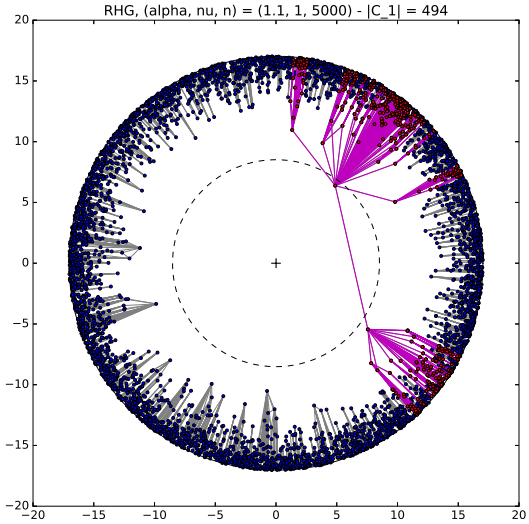
$n = 1000, |C_1| = 512$



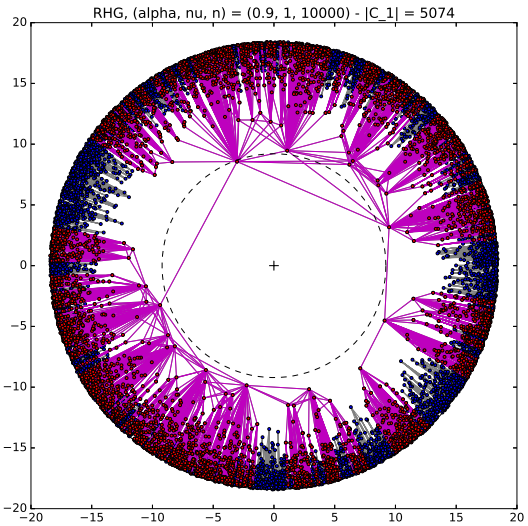
$n = 1000, |C_1| = 189$



$n = 5000, |C_1| = 2638$

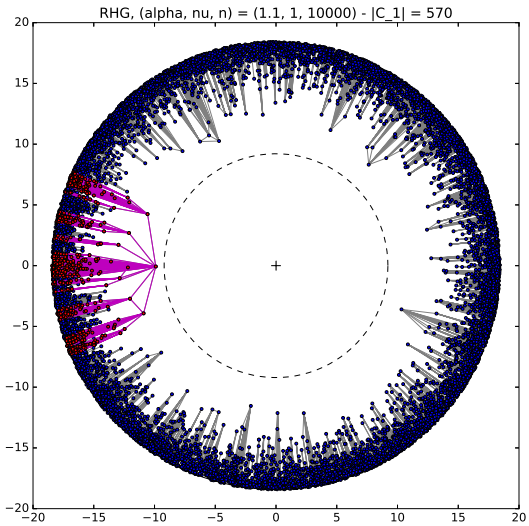


$n = 5000, |C_1| = 494$



$n = 10000, |C_1| = 5074$

(a) $\mathcal{G}_{0.9,\,1}$: for $\alpha < 1$, $|C_1|$ grows linearly



$n = 10000, |C_1| = 570$

(b) $\mathcal{G}_{1.1,\,1}$: for $\alpha > 1$, $|C_1|$ grows sublinearly

Figure 3.8: Illustration of Theorem 2.3. The giant component of each graph is coloured in magenta/red.

When $\alpha < 1$, the theorem affirms that the size of the giant component is at least linear with n, that is to say that fixing $\alpha < 1$ and $\nu > 0$, there exists a constant $c = c(\alpha, \nu)$ such that $|C_1| > cn$ a.a.s.[5]. Figure 8(a) confirms this result: $c = 0.5$ is sufficient.
While when $\alpha > 1$, the size of the component is sub-linear so the ratio $\frac{|C_1|}{n}$ tends to 0 as n tends to $+\infty$. We can observe this in Figure 8(b): when n increases, the size of C_1 is not increasing linearly.
For $\alpha = 1$, we have seen that the behaviour of the giant component depends on ν. It is hard to illustrate this as it requires graphs with a really significant number of vertices to show a real difference.

	$n = 1000$	$n = 5000$	$n = 10000$	$n = 50000$
$\alpha = 0.9$	473.5	2842.4	4832.4	22705.5
$\alpha = 1.1$	174.1	425.4	622.1	1003.0

Figure 3.9: Average size of the giant component of 10 $\mathcal{G}_{\alpha,\,1}(n)$.

In Figure 3.9, we can see the average size of the giant component of 10 random hyperbolic graphs per parameters. This confirms that the growth is *on average* linear when $\alpha > 1$ and sublinear for $\alpha < 1$.

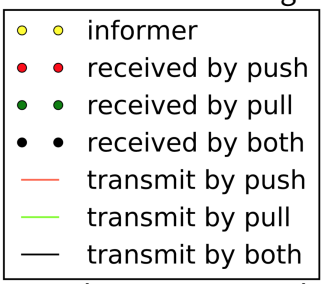
Theorem 2.4 The asymptotical result of this theorem is not really compatible with simulations allowing only a finite number of vertices. Indeed, taking $\alpha < 1/2$, we can still have some isolated vertices when n is fixed. In fact, the distinction between the two cases near to $\alpha = 1/2$ is not clear. Nevertheless, as we have already seen with the figures of this report (see for instance Figure 2.2), a small α induces a connected graph while a bigger α generates a lot of small components.

[5] we recall that C_1 is the giant component of the graph

3.1.4 Push & pull

We ran some simulations of the push & pull rumor spreading into our different models in order to understand the general behaviour and characteristics of the spread.

Let us focus on the legend that we will use for all our simulations:



- the vertex initially informed by the rumor will be in yellow,
- a vertex informed by (a) push(s) of at least one of its neighbours will be in red,
- a vertex informed by a pull will be in green,
- a vertex informed both by (a) push(s) and a pull will be in black,
- an edge used to inform one vertex by a push will be in red,
- an edge used to inform one vertex by a pull will be in green,
- an edge used to inform one vertex both by a push and a pull will be in dark.

Note that a vertex can be informed by more than one of its neighbors, because of pushs: the vertex can make at most one pull per time, but it can receive a push from all its already informed neighbours. This explains the black legend. We focus the reader on the fact that this case only appears when the push and the pull happen at the same time ! Otherwise it is the first process that informed the vertex that counts.

Most of the time, we will just show the giant component (as other vertices will not be informed) and choose uniformly the initially infected vertex.

Simulations into the random hyperbolic graph model In Figures 3.10 and 3.11, we can see two simulations in the same graph. Let us analyse them:

(i) during the first steps, we observe pushs and pulls into the sub-component[6] of the initially informed vertex (with almost only pushs at the beginning),
(ii) the sub-component becomes more and more informed, then we need to transmit the information to the rest of the graph. But most of the time, there are just a few edges available for the transmission. That i's why we have to wait the activation of this edge (see time 14 of Figure 3.10). This gives the information to a vertex with a low radius and thus an important number of connections.
(iii) this important number of connections implies a significant number of pulls during the next steps from edges of the same sub-component but with bigger radii and lower degree.
(iv) the same process (steps (ii) and (iii)) happens while there are some non-informed isolated sub-components.

Obviously, the position of the initially informed vertex is significant:

- a vertex with a big radius will imply step (i), while if it has a small radius the process will directly move on to step (iii),
- a vertex located on one side of the component will take more time to inform the rest of the component, as there will be more phases of propagation.

By comparing Figures 3.10 and 3.11, we can see the effect of the initial position.

[6] we do not give a precise definition of the sub-component, but we can see on the graphs some regions with a lot of internal connections but with one or two connections between the regions. We call those regions the sub-components

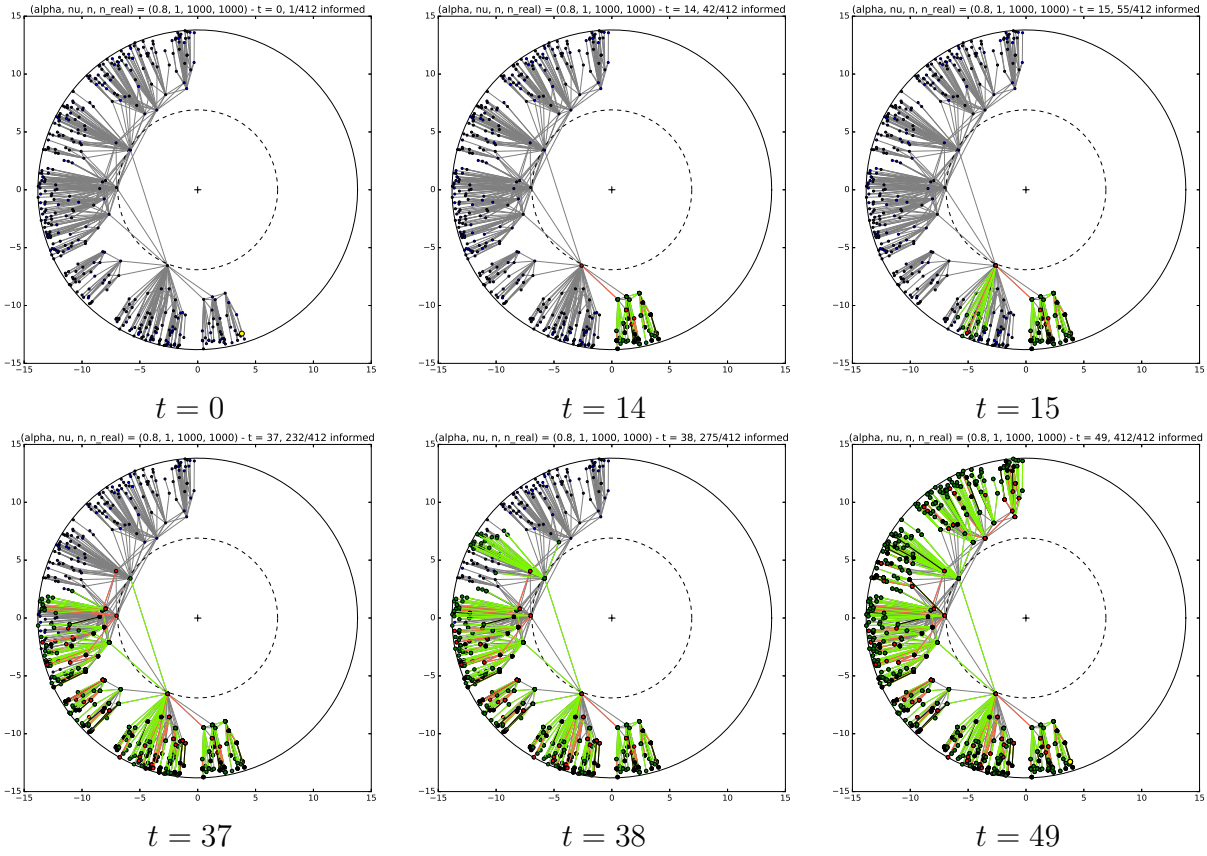


Figure 3.10: A push & pull simulation into $\mathcal{G}_{0.8,\,1}(1000)$.

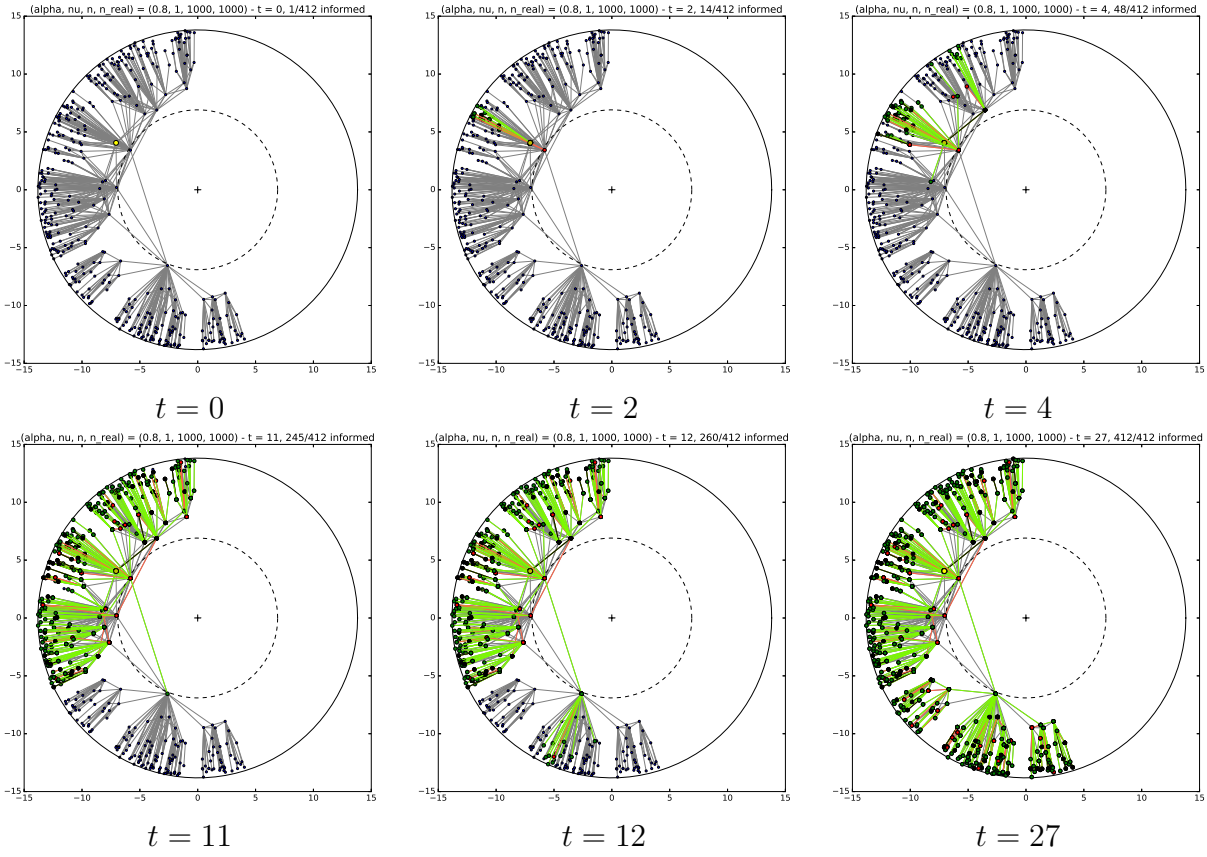


Figure 3.11: Another simulation into the same RHG, starting at another vertex.

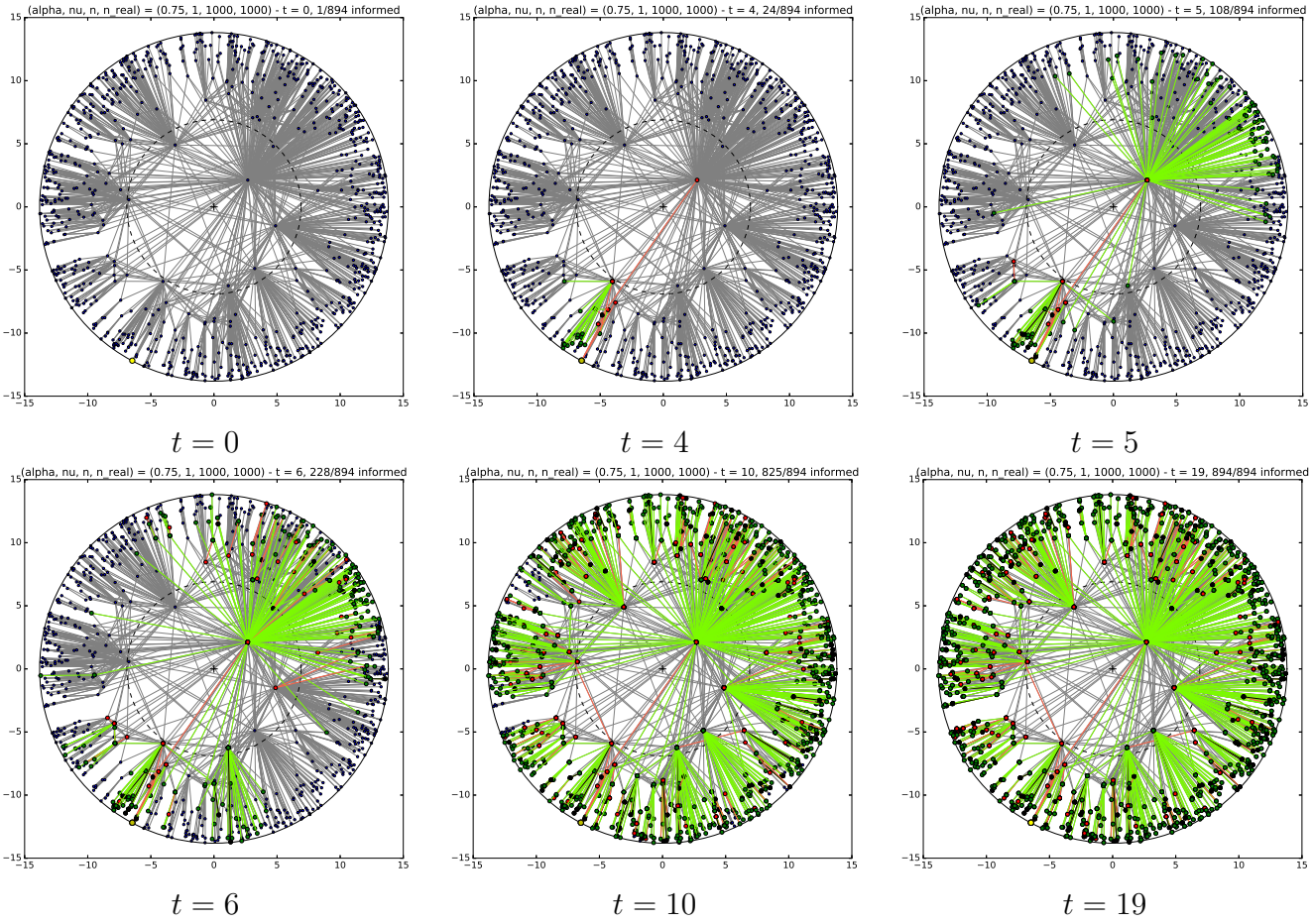
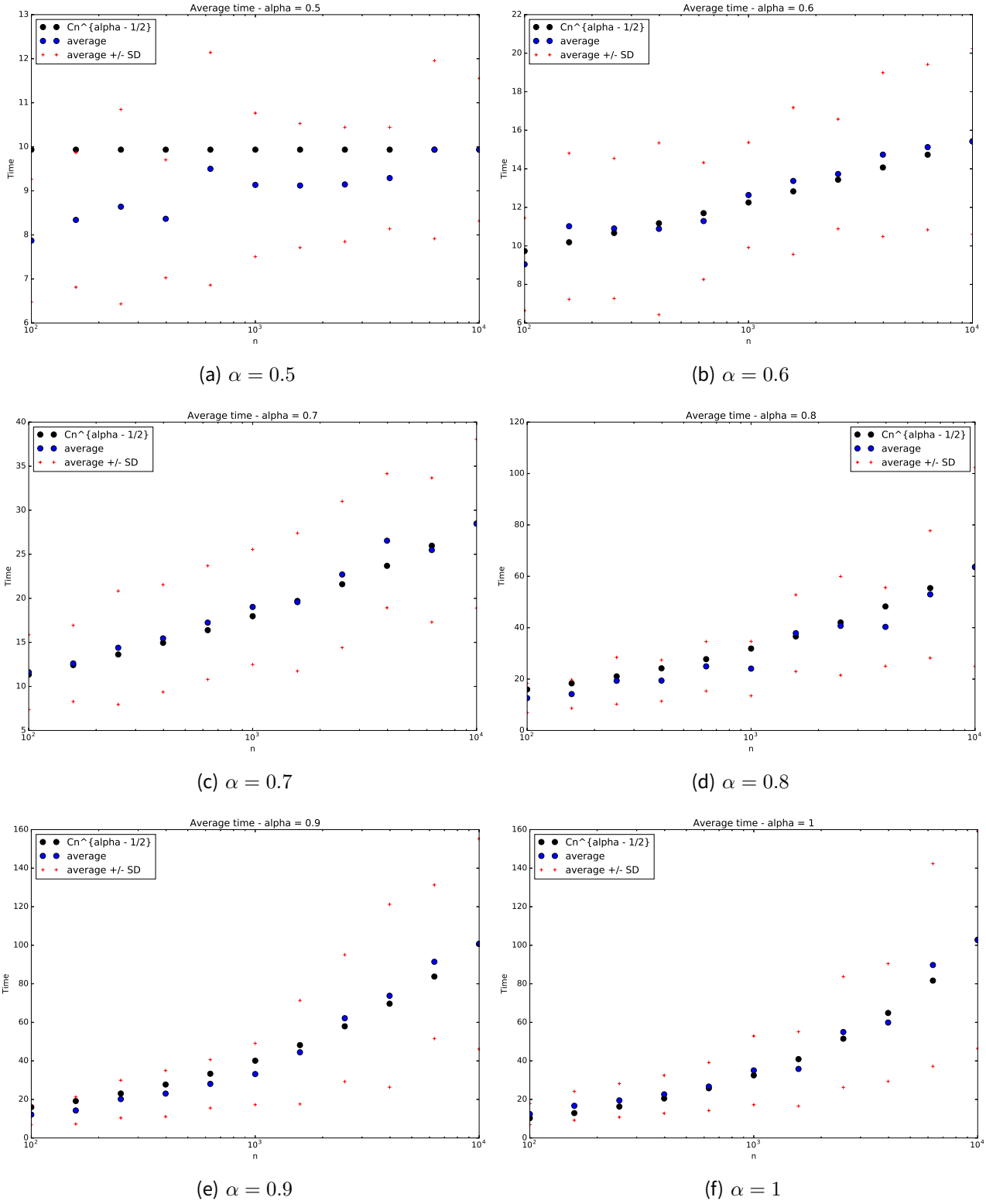


Figure 3.12: A push & pull simulation into $\mathcal{G}_{0.75,\,1}(1000)$.

Figure 3.12 is another simulation in a new graph. First we observe that the giant component is really bigger than the one of the precedent graph, even if α changes just a little.
During the simulation, the process mentioned above seems to happen only one time, as there is no really isolated sub-component. The transmission at time $t = 4$ to a central vertex directly informs all regions of the graph.

Conjecture on the average spreading time In Figure 3.13, we observe the evolution of the average spreading time into $\mathcal{G}_{\alpha,\,1}(n)$. In each subfigure, we fix α and then for some values of n, we generate 20 random hyperbolic graphs in which we realise 10 push & pull simulations. We then show the average time of those simulations in blue, plus or minus the standard deviation for the red curves.
At the sight of those simulations, a plausible conjecture is that the average time is in $\Theta(n^{\alpha-1/2})$ (see the concordance of the blue and black curves). In Paragraph 3.2.2, we try to give theoretical arguments to support the latest conjecture.



(a) $\alpha = 0.5$ (b) $\alpha = 0.6$

(c) $\alpha = 0.7$ (d) $\alpha = 0.8$

(e) $\alpha = 0.9$ (f) $\alpha = 1$

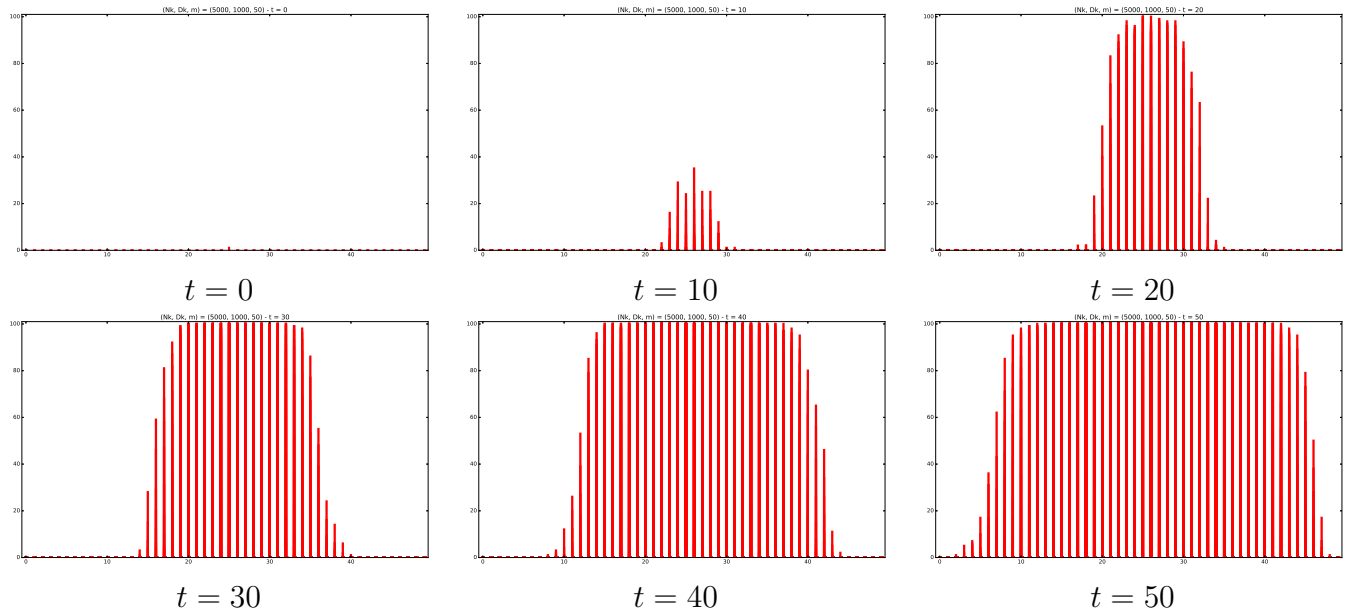
Figure 3.13: Average spreading time into $\mathcal{G}_{\alpha,1}(n)$. For each pair (α, n), we simulate 200 push & pull processes into 20 $\mathcal{G}_{\alpha,1}(n)$ (10 simulations per graph).

Other models In Paragraph 3.2.3, we introduce some graph models that we theoretically study in order to better understand the spread properties. For the sake of conciseness, we do not want to put simulations of all the models in this report. The reader can find them on YouTube.
Nevertheless, we will quickly present a simulation corresponding to the auxiliary graph that we studied most.

Let $m, c \in \mathbb{N}^*$ and $D \geq 3c - 1$. We consider the following graph:

- the set of vertices is composed of m groups of size c (thus there are $N = m \times c$ vertices). Let us denote them by $K_0, K_1, \ldots, K_{m-1}$.
- the set of edges is such that all groups are cliques and all pair of two consecutive groups are inducing a complete bipartite graph. So $K_0 \cup K_1$, $K_1 \cup K_2$, ...and $K_{m-1} \cup K_0$ are cliques.
- we then add loops to all vertices such that they end up with degree *exactly* D (which is possible as $D \geq 3c - 1$, as each vertex is neighbor to the vertices of 3 groups, thus $3c - 1$ vertices).

This graph corresponds to the auxiliary graph $\hat{G}_k$ introduced in Paragraph 3.2.3. Parameters m, c and D are in this graph depending on α, ν, n but we do not want to focus on this for the simulation. We just keep in mind that $m \ll c \ll D$ and all those parameters are big.
We want to understand how the propagation happens. Let us fix a first informed vertex. Without loss of generality, we can assume that this vertex belongs to $K_{m/2}$. In Figure 3.14, we can see a typical spread. In this simulation, we have $c = 100$ vertices per group and we watch the evolution of the number of informed vertices per groups at each time.



$t = 0$ $t = 10$ $t = 20$

$t = 30$ $t = 40$ $t = 50$

Figure 3.14: A propagation into the auxiliary graph with one layer, taking $(N_k, D_k, m) = (5000, 1000, 50)$.

The main observation that we will use in our reasoning (see Paragraph 3.2.5) is the following: there is a kind of wave that propagates more or less at a constant speed, from the moment when at least a few groups are almost fully informed. We will use this constant speed to decompose our steps in phases of same length of time. Moreover, the tail of the wave[7] seems to satisfy the following property: the ratio between the number of informed vertices in two consecutive groups is almost constant. This is another important idea of our reasoning.

[7] groups for which at least one vertex had already receive the information but not all

3.2 Mathematical approach for the push & pull model

In this section, we fix $\alpha \in]1/2, 1[$, $\nu > 0$ and $n \in \mathbb{N}^*$. We set $\mathcal{G} := \mathcal{G}_{\alpha,\nu}(n)$. We recall that $R = 2\log(n/\nu)$ and that $\Delta\theta(r_1, r_2)$ represents the maximal angle formed by two nodes at distance R, at radius r_1 and r_2 respectively from the origin.

We will introduce later some graph models in which vertices can have loops. For the sake of clarity, we will use the term *edge* only for a link between two distincts vertices, while we refer to a *loop* for a link that relies a vertex to itself. We refer to a *link* when considering an edge or a loop without distinction. The degree of a vertex is equal to the number of its links[8].

Here are some notations:

Definition 3.1. In $\mathcal{G}$:

- we denote by Lyr_i the set of vertices of layer i, that is to say the vertices with a radial coordinate in $]i-1, i]$:
$$\mathrm{Lyr}_i := \{v \in V \mid i-1 < r_v \leq i\}$$
- $N_i := \mathbb{E}[\mathrm{card}(\mathrm{Lyr}_i)]$ represents the expected number of vertices in Lyr_i,
- $D_i := \mathbb{E}[\deg(v) \mid v \in \mathrm{Lyr}_i]$ represents the expected degree of a vertex in Lyr_i,
- T represents the time a push & pull spread takes to inform all vertices.

We also define:

Definition 3.2. Let $\mu = \mu_{\alpha,\,\nu}$ be the probability measure according to which the points are generated, that is to say that for every point set S, we have

$$\mu(S) := \int_S f(u)\,du = \frac{1}{2\pi}\int_S \rho(r_u)\,du = \frac{1}{2\pi}\int_S \frac{\alpha\sinh(\alpha r_u)}{\cosh(\alpha R)-1}\mathbb{1}_{[0,R[}(r_u)\,du$$

Note that μ is rotation-invariant as the last integral depends only on the radial coordinate of the points of S.

3.2.1 Useful identities

In this paragraph, we introduce some important lemmas for the study of random hyperbolic graphs.

Lemma 3.1. *Let $r_1, r_2 \in [0, R]$. Then:*

$$\Delta\theta(r_1, r_2) = \begin{cases} \pi & \text{if } r_1 + r_2 \leq R \\ 2\,\mathrm{e}^{\frac{R-r_1-r_2}{2}}(1+\Theta(\mathrm{e}^{R-r_1-r_2})) & \text{otherwise} \end{cases}$$

The case $r_1 + r_2 \leq R$ is a simple application of the triangle inequality. For the other case, we use trigonometric identities and Taylor series to prove this lemma, but no particularly interesting ideas for the model. A complete proof of this result can be found in [GPP12].

[8] note that loops are not counted twice, as it could be the case in the literature

Lemma 3.2. *Let $u = (r_u, \theta_u) \in [0, R] \times [0, 2\pi[$ and $r = r(n) \in [0, R]$ be a function of n such that $r(n) \underset{n\to+\infty}{\longrightarrow} +\infty$. We have:*

$$\mu(\mathbb{B}_{\mathbb{H}}(O, r)) = \mathrm{e}^{-\alpha(R-r)}(1 + o(1)) \tag{4}$$

$$\mu(\mathbb{B}_{\mathbb{H}}(u, R) \cap \mathbb{B}_{\mathbb{H}}(O, R)) = C(\alpha)\,\mathrm{e}^{-r_u/2}(1 \pm \mathcal{O}(\mathrm{e}^{-(\alpha-1/2)r_u} + \mathrm{e}^{-r_u})) \tag{5}$$

where $C(\alpha) = \frac{2\alpha}{\pi(\alpha-1/2)}$.

Further, for $r \leq R - r_u$:

$$\mu(\mathbb{B}_{\mathbb{H}}(u, R) \cap \mathbb{B}_{\mathbb{H}}(O, R) \setminus \mathbb{B}_{\mathbb{H}}(O, r)) = C(\alpha)\,\mathrm{e}^{-r_u/2}(1 \pm \mathcal{O}(\mathrm{e}^{-(\alpha-1/2)r_u} + \mathrm{e}^{-r_u})) \tag{6}$$

while for $r \geq R - r_u$ it holds that:

$$\mu(\mathbb{B}_{\mathbb{H}}(u, R) \cap \mathbb{B}_{\mathbb{H}}(O, R) \setminus \mathbb{B}_{\mathbb{H}}(O, r)) = C(\alpha)\,\mathrm{e}^{-r_u/2}\left(1 - \left(1 + \frac{\alpha - 1/2}{\alpha + 1/2}\,\mathrm{e}^{-2\alpha r}\right)\mathrm{e}^{-(\alpha-1/2)(R-r)}\right)\left(1 \pm \mathcal{O}(\mathrm{e}^{-r_u} + \mathrm{e}^{-r_u-(R-r)(\alpha-3/2)})\right) \tag{7}$$

This lemma introduces useful identities and uses classical integrals calculation for random hyperbolic graphs. This is why we recall the proof of those results:

Proof We have:

$$\begin{aligned}\mu(\mathbb{B}_{\mathbb{H}}(O, r)) &= \frac{1}{2\pi}\int_{\mathbb{B}_{\mathbb{H}}(O,r)} \frac{\alpha\sinh(\alpha r_v)}{\cosh(\alpha R) - 1}\,dv = \int_0^r \frac{\alpha\sinh(\alpha\tilde{r})}{\cosh(\alpha R) - 1}\,d\tilde{r} = \frac{\cosh(\alpha r) - 1}{\cosh(\alpha R) - 1} \\ &= \frac{\mathrm{e}^{\alpha r} + \mathrm{e}^{-\alpha r} - 2}{\mathrm{e}^{\alpha R} + \mathrm{e}^{-\alpha R} - 2} = \mathrm{e}^{-\alpha(R-r)}\,\frac{1 + \mathrm{e}^{-2\alpha r} - 2\,\mathrm{e}^{-\alpha r}}{1 + \mathrm{e}^{-2\alpha R} - 2\,\mathrm{e}^{-\alpha R}} = \mathrm{e}^{-\alpha(R-r)}(1 + o(1))\end{aligned}$$

which gives equation (4).

Let us now prove the other results. For equation (5), we have:

$$\mu(\mathbb{B}_{\mathbb{H}}(u, R) \cap \mathbb{B}_{\mathbb{H}}(O, R)) = \int_0^R \int_{-\Delta\theta(r_u,r)}^{\Delta\theta(r_u,r)} f(r)\,d\theta\,dr = 2\int_0^R \Delta\theta(r_u, r) f(r)\,dr$$

Using Lemma 3.1, it follows that:

$$\begin{aligned}\mu(\mathbb{B}_{\mathbb{H}}(u, R) \cap \mathbb{B}_{\mathbb{H}}(O, R)) &= 2\pi\int_0^{R-r_u} f(r)\,dr + 2\int_{R-r_u}^R \Delta\theta(r_u, r) f(r)\,dr \\ &= \mu(\mathbb{B}_{\mathbb{H}}(O, R - r_u)) + I(R - r_u)\end{aligned}$$

where $I(x) = 2\int_x^R \Delta\theta(r_u, r) f(r)\,dr$. For equation (7), we have for $R - r_u \leq r \leq R$:

$$\mu(\mathbb{B}_{\mathbb{H}}(u, R) \cap \mathbb{B}_{\mathbb{H}}(O, R) \setminus \mathbb{B}_{\mathbb{H}}(O, r)) = 2I(r)$$

Then we must compute $I(x)$ for $x \geq R - r_u$. We use Lemma 3.1 to obtain:

$$\begin{aligned}I(x) &= 2\int_x^R 2\,\mathrm{e}^{\frac{R-r_u-y}{2}}(1 \pm \mathcal{O}(\mathrm{e}^{R-r_u-y}))\frac{\alpha\sinh(\alpha y)}{2\pi(\cosh(\alpha R) - 1)}\,dy \\ &= \frac{2\alpha}{\pi(\cosh(\alpha R) - 1)}\int_{R-r_u}^R \mathrm{e}^{\frac{R-r_u-y}{2}}(1 \pm \mathcal{O}(\mathrm{e}^{R-r_u-y}))\sinh(\alpha y)\,dy\end{aligned}$$

Let us consider first the integral without the error term. Using that $y \longmapsto \frac{2}{4\alpha^2-1}\,\mathrm{e}^{-y/2}(2\alpha\cosh(\alpha y) + \sinh(\alpha y))$ is a primitive of $y \longmapsto \mathrm{e}^{-y/2}\sinh(\alpha y)$, we obtain:

$$\int_x^R \mathrm{e}^{\frac{R-r_u-r}{2}}\sinh(\alpha r)\,dr = \frac{2\,\mathrm{e}^{\frac{-r_u}{2}}\left[(2\alpha\cosh(\alpha R) + \sinh(\alpha R)) - \mathrm{e}^{\frac{R-x}{2}}(2\alpha\cosh(\alpha x) + \sinh(\alpha x))\right]}{4\alpha^2 - 1}$$

Then as $(\cosh(\alpha R)-1)^{-1} = 2\,\mathrm{e}^{-\alpha R}(1+\Theta(\mathrm{e}^{-\alpha R}))$, one can obtain:

$$\int_x^R \mathrm{e}^{\frac{R-r_u-r}{2}} \sinh(\alpha r)\,dr = \frac{\alpha\,\mathrm{e}^{-r_u/2}}{\alpha-1/2}\left(1-\left(1+\frac{\alpha-1/2}{\alpha+1/2}\right)\mathrm{e}^{-(\alpha-1/2)(R-x)}\right)(1+\Theta(\mathrm{e}^{-\alpha R}))$$

We consider now the error term. Recall that $x \geq R-r_u$, we have:

$$\int_x^R \mathrm{e}^{\frac{R-r_u-y}{2}}\,\mathcal{O}(\mathrm{e}^{R-r_u-y})\frac{\sinh(\alpha y)}{\cosh(\alpha R)-1}\,dy = \int_x^R \mathcal{O}(\mathrm{e}^{\frac{3}{2}(R-r_u-y)+\alpha(y-R)})\,dy = \mathcal{O}\left(\mathrm{e}^{-3r_u/2}+\mathrm{e}^{-3r_u/2-(R-x)(\alpha-3/2)}\right)$$

Finally, it follows that:

$$I(x) = C(\alpha)\,\mathrm{e}^{-r_u/2}\left(1-\left(1+\frac{\alpha-1/2}{\alpha+1/2}\right)\mathrm{e}^{-(\alpha-1/2)(R-x)}\right)\left(1\pm\mathcal{O}(\mathrm{e}^{-r_u}+\mathrm{e}^{-r_u-(R-x)(\alpha-3/2)})\right) \tag{8}$$

This gives equations (7). We can also deduce equation (5), taking $x = R-r_u$:

$$\mu(\mathbb{B}_{\mathbb{H}}(u,R)\cap\mathbb{B}_{\mathbb{H}}(O,R)) = \mathrm{e}^{-\alpha r_u}(1+o(1)) + C(\alpha)\,\mathrm{e}^{-r_u/2}\left(1\pm\mathcal{O}(\mathrm{e}^{-r_u}+\mathrm{e}^{-(\alpha-1/2)r_u})\right)$$

and as $\mathrm{e}^{-\alpha r_u} = \mathrm{e}^{-r_u/2-(\alpha-1/2)r_u}$, we can put the first term into the error of the second, which gives the result.

Equation (6) can be shown in a similar way. □

From this lemma, we deduce two important results:

Lemma 3.3. *Let $i = i(n) \in [1,R]$ be a layer such that $i(n) \underset{n\to+\infty}{\longrightarrow} +\infty$. Then:*

$$N_i = n(1-\mathrm{e}^{-\alpha})\,\mathrm{e}^{-\alpha(R-i)}(1+o(1))$$

and

$$D_i = C(\alpha)\,\mathrm{e}^{\frac{1}{2}(R-i)}(1+o(1))$$

Proof By independence of the vertices, we have:

$$\begin{aligned} N_i := \mathbb{E}[\mathrm{card}(\mathrm{Lyr}_i)] &= n\mu(\mathbb{B}_{\mathbb{H}}(O,i)\setminus\mathbb{B}_{\mathbb{H}}(O,i-1)) \\ &= n\left[\mathrm{e}^{-\alpha(R-i)}(1+o(1)) - \mathrm{e}^{-\alpha(R-(i-1))}(1+o(1))\right] \\ &= n(1-\mathrm{e}^{-\alpha})\,\mathrm{e}^{-\alpha(R-i)}(1+o(1)) \end{aligned}$$

Assuming that there is a vertex u on layer i, the set of its neighbours is the set of vertices of $V\setminus\{u\}$ that belongs to $\mathbb{B}_{\mathbb{H}}(u,R)$. Thus:

$$\begin{aligned} D_i := \mathbb{E}[\deg(v)\,|\,v\in\mathrm{Lyr}_i] &= (n-1)\mu(\mathbb{B}_{\mathbb{H}}(u,R)\cap\mathbb{B}_{\mathbb{H}}(O,R)) \\ &= (n-1)C(\alpha)\,\mathrm{e}^{-i/2}(1\pm\mathcal{O}(\mathrm{e}^{-(\alpha-1/2)i}+\mathrm{e}^{-i})) \\ &= C(\alpha)\,\mathrm{e}^{\frac{1}{2}(R-i)}(1\pm o(1)) \end{aligned}$$

□

3.2.2 About the conjecture

We have seen with the push & pull simulations (Paragraph 3.1.4) that the average spreading time might be of order $\Theta(n^{\alpha-1/2})$. In this short paragraph we explain our theoretical intuition behind this conjecture. This is obviously not a proof.

A typical spread starts by informing a vertex of big radius, near to the boundary. In a really quick time, most of the vertices of its sub-component will become informed, including the one with the lowest radius. We think that the time of this first phase is not far from being constant (that is to say not depending on n).
But in order to spread the rumor to the other sub-components, only a few edges can be used. Those edges connect vertices of small radius. Suppose those vertices are in $\mathbb{B}(O, R/2)$. Then the average time taken by the informed vertex to inform another vertex of the ball is:

$$\begin{aligned}\tau &= \mathbb{P}(\text{inform at least one vertex of } \mathbb{B}(O,R/2) \text{ in one step})^{-1} \\ &\simeq \Big(\mathbb{P}(\text{inform a fixed vertex}) \times \mathrm{card}(\{v \in \mathbb{B}(O,R/2)\})\Big)^{-1} \qquad \text{(union bound is a good approximation)} \\ &\simeq \left(\frac{2}{D_{R/2}} \times N_{R/2}\right)^{-1}\end{aligned}$$

This last term is proportional to $n^{\alpha-1/2}$. Thus the second phase consisting to inform a few vertices of $\mathbb{B}(O, R/2)$ takes approximately some $\Theta(n^{\alpha-1/2})$ steps. Then all other vertices of the graph are quickly informed (using pulls) in a third phase, almost constant.

Therefore the longest phase is the one in $\Theta(n^{\alpha-1/2})$.

3.2.3 Useful graph models

In order to simplify the analysis we propose some models of graphs. The goal is to understand how the spread happens into one layer or between two layers of $\mathcal{G}$. We fix $\nu = 1$ in order to avoid some constant factors and be more readable.

First model: the random hyperbolic graph in expectation We consider the following graph, denoted[9] by G:

- the set of vertices V is divided into layers: $V = \cup_{1 \le i \le \lfloor R \rfloor} \mathrm{Lyr}_i$. Each layer Lyr_i is a set of vertices such that $\mathrm{card}(\mathrm{Lyr}_i) = N_i$ and vertices are arranged uniformly on the circle of radius i, that is to say that their radius is equal to i and the angle between two consecutive vertices is $2\pi/N_i$,
- the set of edges E is then generated with the same rule as in random hyperbolic graphs: we put an edge between two vertices if their hyperbolic distance is less or equal to R,
- we then add loops to vertices of each layer Lyr_i so they end up with degree[10] D_i.

Second model: the one layer graph Let $R/2 \le k \le R$. We consider the following graph, denoted by G_k :

- the set of vertices $V = \mathrm{Lyr}_k$ is made up of vertices into only one layer, where Lyr_k is generated as in the last graph,
- the set of edges E is then generated with the same rule as random hyperbolic graphs,
- we then add loops to all vertices so they end up with degree *exactly* D_k.

[9] remark that G and the other considered graphs still depend on α, ν, n

[10] for this model we do not justify that there can not be some vertices in Lyr_i with more than D_i edges. If it is not the case we do not add loops to those vertices

We have to justify that after the generation of E, we do not have vertices with more than D_k edges. Let D_k^{in} denote the number of neighbors a vertex in Lyr_k has within its same layer. Only vertices at angular distance at most $\Delta\theta(k,k)$ are neighbors to a fixed vertex, so it represents only a $2\Delta\theta(k,k)/(2\pi)$ fraction of the vertices of Lyr_k. Using Lemmas 3.1 and 3.3, we have :

$$\begin{aligned} D_k^{in} &= N_k \frac{2\Delta\theta(k,k)}{2\pi} = \frac{1}{\pi} n(1-\mathrm{e}^{-\alpha})\,\mathrm{e}^{-\alpha(R-k)}\,2\,\mathrm{e}^{\frac{R-2k}{2}}(1+o(1))(1+\Theta(\mathrm{e}^{R-2k})) \\ &= \frac{2}{\pi}(1-\mathrm{e}^{-\alpha})\,\mathrm{e}^{R/2}\,\mathrm{e}^{-\alpha(R-k)}\,\mathrm{e}^{\frac{R-2k}{2}}(1+o(1)) = \frac{2}{\pi}(1-\mathrm{e}^{-\alpha})\,\mathrm{e}^{(1-\alpha)(R-k)}(1+o(1)) \end{aligned}$$

Thus $D_k^{in} = o(D_k)$ (Lemma 3.3). Then for a large enough n, we do not exceed degree D_k before adding loops for any vertex.

Third model: the two layers graph Let $R/2 \leq a < b \leq R$. We consider the following graph, denoted by $G_{a,b}$:

- the set of vertices V is divided into two layers: we have $V = \mathrm{Lyr}_a \cup \mathrm{Lyr}_b$, still generated similarly,
- the set of edges E is then generated with the same rule as random hyperbolic graphs,
- we then add loops to vertices of each layer Lyr_i so they end up with degree exactly D_i.

We have to make the same justification as in the last model. Let D_a^{ext} (resp. D_b^{ext}) denote the number of neighbors a vertex in Lyr_a (resp. Lyr_b) has outside its layer. We have:

$$\begin{aligned} D_a^{ext} &= N_b \frac{2\Delta\theta(a,b)}{2\pi} = \frac{2}{\pi}(1-\mathrm{e}^{-\alpha})\,\mathrm{e}^{R/2}\,\mathrm{e}^{-\alpha(R-b)}\,\mathrm{e}^{\frac{R-a-b}{2}}(1+o(1)) \\ &= \frac{2}{\pi}\,\mathrm{e}^{\frac{R-a}{2}}\,\mathrm{e}^{-(\alpha-1/2)(R-b)}(1+o(1)) \\ D_b^{ext} &= N_a \frac{2\Delta\theta(a,b)}{2\pi} = \frac{2}{\pi}\,\mathrm{e}^{\frac{R-b}{2}}\,\mathrm{e}^{-(\alpha-1/2)(R-a)}(1+o(1)) \end{aligned}$$

One can check that $D_a^{ext} = o(D_a)$ and $D_b^{ext} = o(D_b)$. Thus $D_a^{in} + D_a^{ext} = o(D_a)$ and $D_b^{in} + D_b^{ext} = o(D_b)$ using the calculation of the last model. For n large enough, all vertices have a lower degree than expected before adding the loops.

Two auxiliary graphs Let $R/2 \leq k \leq R$. We consider the following graph, denoted by $\hat{G}_k$:

- we have $V = \mathrm{Lyr}_k$ as in the one layer graph. Vertices of Lyr_k are partitioned into groups $K_0, \ldots, K_{m_k-1}$ of consecutive (in angular distance) vertices so that each group K_i spans a set of vertices at angular distance at most $\Delta\theta(k,k)$. We refer to a K_i as a K-group and as all groups have same length, we put $|K| = |K_i|$. We have $m_k = N_k/|K|$.
- the set of edges E is then generated such that each pair of two consecutive K-groups induces a clique[11].
- we then add loops to all vertices so they end up with degree exactly D_k.

Note (and check) that each edge of the one layer graph is an edge of this graph. It is easy to see[12] that $D_k^{in}(\hat{G}_k) = \frac{3}{2} D_k^{in}(G_k)$ and so $D_k^{in}(\hat{G}_k) = o(D_k)$.

Let $\frac{R}{2} \leq a < b \leq R$. We consider the following graph, denoted by $\hat{G}_{a,b}$:

- we have $V = \mathrm{Lyr}_a \cup \mathrm{Lyr}_b$ as in the two layer graph. Similarly as in $\hat{G}_k$, vertices of Lyr_a (resp. Lyr_b) are partitioned into groups $A_0, \ldots, A_{m_a-1}$ (resp. $B_0, \ldots, B_{m_b-1}$) with associated angular distance $\Delta\theta(a,a)$ (resp. $\Delta\theta(b,b)$),

[11] so $K_0 \cup K_1$, $K_1 \cup K_2$, ... and $K_{m-1} \cup K_0$ are cliques

[12] indeed $D_k^{in}(\hat{G}_k) = 3\,|K| = 3N_k\Delta\theta(k,k) = \frac{3}{2} D_k^{in}(G_k)$

- in each layer, the set of edges E is then generated as in $\hat{G}_{a,b}$. We then connect a vertex of Lyr_a and a vertex of Lyr_b according to the rule of random hyperbolic graphs.
- we then add loops to vertices of each layer Lyr_i so they end up with degree exactly D_i.

There again, we do not exceed the expected degree of each vertex (all arguments have already been mentioned earlier).

Goals We think that the spread will have the same properties in $\mathcal{G}$ and G. The understanding of what happens in G might be easier.
But there are many layers in G. That's why we introduce G_k and $G_{a,b}$: the first objective is to understand the spread in one layer, and then watch what happens when we add another layer: does this layer help the spread into the first layer or not? The behaviour could depend on the value of b: a layer close to a might not influence similarly as a layer close to the boundary R and thus far from a.
As we will see in Lemma 3.4, the push & pull will process faster in the auxiliary graphs. They are introduced to find a lower bound for the one layer and the two layer graphs. We thought that their study would be easier than the one layer and the two layer graphs, but we will see later that it is still far from being simple.

3.2.4 Comparison of the push & pull spread time in graphs

LEMMA 3.4. [PUSH & PULL IS FASTER IN THE AUXILIARY GRAPH]

We have:

$$T(\hat{G}_k) \lesssim T(G_k) \quad \textit{and} \quad T(\hat{G}_{a,b}) \lesssim T(G_{a,b})$$

PROOF Let us demonstrate the result for the one layer case. We will provide a coupling such that the push & pull propagation is always faster in $\hat{G}_k$ than in G_k. We can adapt this proof to the two layer case, using exactly the same arguments.

We assume n large enough such that all vertices have loops in $\hat{G}_k$.
We write $G_k = (V, E)$ and $\hat{G}_k = (V, \hat{E})$.

Let us fix a vertex $u \in V$. We consider $N_u = \{v \mid \{u, v\} \in E\}$ and $\hat{N}_u = \{v \mid \{u, v\} \in \hat{E}\}$. It is easy to see that $N_u \subseteq \hat{N}_u$ as each edge of G_k is an edge of $\hat{G}_k$.

In G_k, u has $D_k - |N_u|$ loops while it has $D_k - |\hat{N}_u|$ in $\hat{G}_k$. To each vertex v of $\hat{N}_u \setminus N_u$, we associate a loop, called l_v, of u in G_k. See that non-associated loops of u in G_k are in same number as the loops of u in $\hat{G}_k$, thus we say they are the same loops.

We consider the following coupling for the push & pull process into the two graphs: when a fixed vertex u uses one of its links in G_k to try to propagate the information:

- if this link was an edge in G_k, the same edge is used in $\hat{G}_k$,
- if this link was a loop:
  - if this loop is one of the $\{l_v\}_{v \in \hat{N}_u \setminus N_u}$, the edge between u and the associated v is used in $\hat{G}_k$,
  - if it is another loop, we use the same loop in $\hat{G}_k$.

Denote by $I_t = \{v \mid v \text{ is informed at } t \text{ in } G_k\}$ (resp. $\hat{I}_t = \{v \mid v \text{ is informed at } t \text{ in } \hat{G}_k\}$) the set of informed vertices at time t in G_k (resp $\hat{G}_k$), we will show by induction that according to this coupling $I_t \subseteq \hat{I}_t$ at all times.

- At time $t = 0$, the two sets are equals (they contain the first informed vertex).
- We now assume that at time $t \geq 0$, we have $I_t \subseteq \hat{I}_t$.
  Consider a vertex u in $I_{t+1} \setminus I_t$. There exists a vertex $v \in I_t$ such that u or v tries to use the edge

$\{u, v\}$ to propagate the information in G_k at time $t+1$. Then by the coupling, this edge has also been used in $\hat{G}_k$. So $u \in \hat{I}_{t+1}$. Thus $I_{t+1} \subseteq \hat{I}_{t+1}$.

To conclude, let us notice that $T(G_k) = \min(\{t \geq 0 \mid I_t = V\})$ and $T(\hat{G}_k) = \min(\{t \geq 0 \mid \hat{I}_t = V\})$. Using the coupling, we have that $\forall \omega, T(\hat{G}_k)(\omega) \leq T(G_k)(\omega)$. □

3.2.5 Study of $\hat{G}_k$

In this paragraph, we present some ideas that we tried to develop to find bounds on $\mathbb{E}[T(\hat{G}_k)]$. The main ideas have been observed in our simulations (see the end of Paragraph 3.1.4). We do not claim to give a precise proof, but we hope that those ideas will be specified soon. We will only consider the push case[13] in order to simplify the study.

We recall that the typical behaviour observed is the following: there is a kind of wave that propagates more or less at a constant speed, from the moment when at least a few groups are almost fully informed. Moreover, the tail of the wave seems to satisfy the following property: the ratio between the number of informed vertices in two consecutive groups is almost constant.

Without loss of generality we assume that the rumor starts at a vertex in group K_0. Sub-indices of the K-groups are modulo m_k, so $K_{m_k - j} = K_{-j}$ ... and we consider most of the time indices between $-m_k/2$ and $m_k/2$, especially when indices are used as a power.
Moreover, let τ_k represent the expected time it takes a vertex in Lyr_k to choose one of its neighbors in Lyr_k (and so pushes the rumor to this neighbor if it does not have it already). We have:

$$\tau_k = \frac{D_k}{3\,|K| - 1}$$

Upper bound We divide the steps into consecutive phases of length Δ (that will be specified shortly). We would like to argue that for an appropriately chosen ρ the following holds: after $r \in \mathbb{N}$ phases, $r < m_k/2$, we have with high probability:

- for $|i| \geq r+1$, then no vertex of K_i is informed,
- for $|i| \leq r$, then at least $\min(\rho^{r-|i|}, |K|)$ vertices in K_i are informed.

Let $I_r(K_i)$ be the set of informed vertices in K_i at the end of the r-th phase.

Consider Δ' the average time needed for a fixed informed vertex in a K-group to inform one vertex of the following K-group, assuming any vertex of this next group has not the information. We have:

$$\Delta' = \mathbb{P}(\text{inform at least one of } |K| \text{ vertices with one vertex})^{-1} \simeq \left(\frac{|K|}{D_k}\right)^{-1} = \frac{D_k}{|K|}$$

where the $\simeq$ is an union bound approximation (that we think will be good as $|K|/D_k \underset{n\to+\infty}{\longrightarrow} 0$).

We set $\Delta = (1+\eta)\Delta'$ for an $\eta > 0$ to be chosen later.
The claim clearly holds for $r = 0$. Assume it holds for r. By symmetry, it is enough to establish that the claim holds for $r+1$ when i is positive.

We fix an i and consider groups K_{i-1} and K_i such that $I_r(K_{i-1}) > 0$ and $I_r(K_i) \leq |K|$. We have $I_r(K_{i-1}) \geq \rho^{r-(i-1)}$ and $I_r(K_i) \geq \rho^{r-i}$. We will see that with high probability, during the $(r+1)$-th phase, the informed vertices of K_{i-1} will inform at least $\rho^{(r+1)-i}$ vertices of K_i.

Indeed, assuming no vertex of K_i is informed (it is not the case by hypothesis, but taking off informed

[13] we consider the same rumor spreading without any pull

vertices only slows the spread), we have if $I_r(K_i) = \rho^{r-i}$ (if it is superior the process is accelerated):

$$\mathbb{E}[\text{number of informed vertices of } K_i \text{ by vertices of } K_{i-1} \text{ during phase } r+1] \simeq \frac{|K|}{D_k}\rho^{r-(i-1)}$$

(still using an union bound).

Thus informing $\rho^{(r+1)-i}$ vertices takes approximately

$$\rho^{(r+1)-i} \times \left(\frac{|K|}{D_k}\rho^{r-(i-1)}\right)^{-1} = \Delta'$$

steps. Using a concentration inequality, we hope to show that during the $r+1$-th phase (of length $\Delta > \Delta'$ steps), with high probability the informed vertices of K_{i-1} inform at least $\rho^{(r+1)-i}$ vertices of K_i for an appropriate $\eta > 0$.

Reasoning similarly with all i will then give the claim for $r+1$.

This argumentation makes believe that $\underline{\mathbb{E}[T(\hat{G}_k)] = \mathcal{O}(m_k D_k / |K|)}$.

Lower bound Let $\varepsilon > 0$ and $\rho := \rho(\varepsilon) > 1$ be a constant, that we will specify later. We will assume that initially, there are $\left\lfloor \rho^{-|i|} |K| \right\rfloor$ informed vertices in K_i for all i. This will make the rumor spreads faster, thing that we can do in the lower bound case. In particular, all vertices in K_0 are informed and no vertex in K_ℓ is informed if and only if $\rho^\ell > |K|$ (clearly $\ell = \left\lceil \frac{1}{\ln(\rho)} \right\rceil \ln(|K|)$).

We divide time into consecutive phases of length $\Delta = \left\lfloor \frac{\tau_k}{C\rho} \right\rfloor (1-\varepsilon)$ for $C > 1$ to be chosen later on. We would like to argue that for an appropriately chosen ρ the following holds: after $r \in \mathbb{N}$ phases, $r < m_k - \ell$, we have with high probability:

- for $|i| \geq r + \ell$, then no vertex of K_i is informed,
- for $r < |i| < r + \ell$, then at most $\rho^{r-|i|} |K|$ vertices in K_i are informed.

Let $N_{r,i} = \rho^{r-|i|} |K|$ be the maximum number of informed vertices in K_i after r phases if the above scenario happens. We recall that $I_r(K_i)$ is the set of informed vertices in K_i at the end of the r-th phase. We want to show that with high probability we have $|I_r(K_i)| \leq N_{r,i}$ for all r and i.

The claim clearly holds for $r = 0$. Assume it holds for r. By symmetry, it is enough to establish that the claim holds for $r+1$ when i is positive. Let $\mathcal{E}$ be the event that for some $i \in [\![r, r+\ell]\!]$, we have $|I_{r+1}(K_i)| > N_{r+1,i}$. Furthermore, let $\mathcal{E}_{i,j}$ for some $r < i < j \leq r + \ell$ be the event that at the end of the $(r+1)$-th phase:

- if $0 \leq s < i$, then $|I_{r+1}(K_s)| \leq N_{r+1,s}$ (at most $\rho^{r+1-s} |K|$ of the vertices in K_s are informed[14]),
- if $i \leq s < j$, then $|I_{r+1}(K_s)| > N_{r+1,s}$ (more than $\rho^{r+1-s} |K|$ vertices in K_s are informed).
- $|I_{r+1}(K_j)| \leq N_{r+1,j}$ (at most $\rho^{r+1-j} |K|$ of the vertices in K_j are informed).

Observe that

$$\mathcal{E} = \bigcup_{i=r+1}^{r+\ell-1} \bigcup_{j=i+1}^{r+\ell} \mathcal{E}_{i,j}$$

It will suffice for our purposes to show that each $\mathcal{E}_{i,j}$ occurs with small probability. We next establish this for the case $j = i+1$ (in fact the case $j > i+1$ will be similar and use the same argument). Let:

- X_i^- be the number of times an informed vertex in K_{i-1} pushes the rumor to a vertex in K_i during the $(r+1)$-th phase.
- X_i^0 be the number of times an informed vertex in K_i pushes the rumor to a vertex in K_i during the $(r+1)$-th phase.

[14] This holds trivially for $0 \leq s \leq r$

- X_i^+ be the number of times an informed vertex in K_{i+1} pushes the rumor to a vertex in K_i during the $(r+1)$-th phase.

We will fix C such that $\frac{1}{C\rho}(\rho^2+\rho+1) = \rho - 1$. Hence, $C = 1 + \frac{\rho^2+2}{\rho-1} > 1$ (see the reason just below).

Note that if $\mathcal{E}_{i,i+1}$ occurs, then either $\mathcal{E}_i^- := \left\{X_i^- > \frac{1}{C\rho}N_{r+1,i-1}\right\}$ or $\mathcal{E}_i^0 := \left\{X_i^0 > \frac{1}{C\rho}N_{r+1,i}\right\}$ or $\mathcal{E}_i^+ := \left\{X_i^+ > \frac{1}{C\rho}N_{r+1,i+1}\right\}$ occurs, since otherwise, at the end of the $(r+1)$-th phase we have:

$$\begin{aligned}
|I_{r+1}(K_i)| &\leq N_{r,i} + X_i^- + X_i^0 + X_i^+ \\
&\leq N_{r,i} + \frac{1}{C\rho}(N_{r+1,i-1} + N_{r+1,i} + N_{r+1,i+1}) \\
&\leq \rho^{r-i}\,|K| + \frac{1}{C\rho}(\rho^{r+2-i} + \rho^{r+1-i} + \rho^{r-i})\,|K| \\
&\leq \rho^{r-i}\,|K| + (\rho-1)\rho^{r-i}\,|K| \\
&\leq \rho^{r+1-i}\,|K| = N_{r+1,i}
\end{aligned}$$

contradicting the fact that there are more than $N_{r+1,i} = \rho^{r+1-i}\frac{D_k}{\tau_k}$ informed vertices in K_i by the end of the $(r+1)$-th phase. We claim that each of the events $\mathcal{E}_i^-$, $\mathcal{E}_i^0$ and $\mathcal{E}_i^+$ occurs with small probability. The arguments should be similar for bounding the probability that each event occurs. Below, we only discuss the bound for the first event.

Recall that by induction hypothesis, $|I_r(K_{i-1})| \leq N_{r,i-1}$. Let T_n be the smallest time step when at least n vertices in K_{i-1} have pushed the rumor towards a vertex in K_i during the $(r+1)$-th phase. Note that it might be the case that $T_n = T_{n+1}$ for some values of n (indeed, it might happened that in just one step the number of informed nodes in K_{i-1} goes from being less than n to being at least $n+1$). Observe that if $\mathcal{E}_i^-$ occurs, then

$$\sum_{0\leq n<\frac{1}{C\rho}N_{r+1,i-1}} T_{n+1} - T_n \leq \left\lfloor \frac{\tau_k}{C\rho}(1-\varepsilon) \right\rfloor$$

Unfortunately, the distribution of $\{T_{n+1} - T_n\}_n$ is not easy to handle, so we further consider that each time step t is subdivided into *substeps*, as many substeps as the number of informed vertices in K_{i-1} at the start of step t (so different t's might be subdivided into different number of substeps). Now, at the start of a time step, all the vertices in K_{i-1} that are informed are arbitrarily ordered and then each one performs a push in that exact order. Let t_n be the substep (stopping time) at which the n-th vertex of K_{i-1} becomes informed.

Clearly, $\{t_{n+1} - t_n\}_{n\in\mathbb{N}}$ are independently distributed. Moreover, $t_{n+1} - t_n$ is distributed according to a geometric distribution with success parameter $\frac{D_k^{in}}{D_k} = \frac{1}{\tau_k}$. By definition of $\mathcal{E}_i^-$, at the end of the $(r+1)$-th phase, the number of informed vertices in K_{i-1} is at most $N_{r+1,i-1} = \rho N_{r,i-1}$, so

$$T_{n+1} - T_n \geq \frac{1}{\rho N_{r,i-1}}(t_{n+1} - t_n)$$

Putting all of our discussion above together, recalling again that $N_{r+1,i-1} = \rho N_{r,i-1}$, we see that if $\{G_n\}_{0\leq n<\frac{1}{C\rho}N_{r+1,i-1}}$ is a family of independent geometric random variables with success probability $\frac{1}{\tau_k}$, then:

$$\mathbb{P}(\mathcal{E}_i^-) \leq \mathbb{P}\left(\frac{1}{\rho N_{r,i-1}} \sum_{0\leq n<\frac{1}{C\rho}N_{r+1,i-1}} G_n \leq \frac{\tau_k}{\rho C}(1-\varepsilon)\right)$$

Note that

$$\mathbb{E}\left[\frac{1}{\rho N_{r,i-1}} \sum_{0\leq n<\frac{1}{C\rho}N_{r+1,i-1}} G_n\right] = \frac{1}{\rho N_{r,i-1}} \sum_{0\leq n<\frac{1}{C\rho}N_{r+1,i-1}} \underbrace{\mathbb{E}[G_n]}_{\tau_k} = \frac{\tau_k}{\rho C}$$

So, as long as $N_{r,i-1}$ is large, then a tail bound on the sum of geometric distributions should give us what we want. For small values of $N_{r,i-1}$, maybe direct calculations would give us a reasonable bound.

There is still one event that we can not control: at the end of the $(r+1)$-th phase, there could be an informed vertex in a K_i-group with $i \geq r+1+\ell$, even if $\mathcal{E}$ does not occur. We do not know how to deal with it.

Forgetting this technical issue, this argumentation let us think that $\underline{\mathbb{E}[T(\hat{G}_k)] = \Omega(m_k D_k / |K|)}$. Maybe a logarithmic factor should be added to obtain the good bound.

Conclusion We hope that $\boxed{\mathbb{E}[T(\hat{G}_k)] = \Theta(m_k D_k / |K|)}$. There is still some work to perform in order to prove this.

3.2.6 Study of $\hat{G}_{a,b}$

In $\hat{G}_{a,b}$, we want to see if Lyr_b is helping Lyr_a to propagate the rumor in the graph, that is to say if $T(\hat{G}_{a,b})$ is less than $T(\hat{G}_a)$. In this short paragraph, we will juste give our intuition, as the ideas we used are quite technical.

Vertices in A_i are partitioned into sub-groups $A'_{im'_a}, \dots, A'_{(i+1)m'_a-1}$, of consecutive (in angular distance) vertices so that each sub-group spans a non-empty set of vertices within angular distance at most $\Delta\theta(a,b)$.

Let us assume that a vertex of one A'-group, say $A' := A'_j$, is informed. We wonder how this vertex will inform a vertex of $A'_+ = A'_{j+1}$ or $A'_- = A'_{j-1}$. There are two strategies:

i) the considered vertex directly pushes the rumor to a vertex of A'_+ or A'_-,
ii) the considered vertex pushes the rumor to a vertex of Lyr_b. Then some pushs happen in Lyr_b, they move the rumor forward or backward in angular distance. At some time, a vertex of Lyr_b which is neighbor of a vertex of A'_+ or A'_-, will be informed and will then push the rumor to A'_+ or A'_-.

The first strategy will take τ_a steps in expectation. The second is less clear and depends on the value of b. It will take about $D_a/|B| = D_a/(\Delta\theta(b,b)N_b)$ steps to inform one vertex of Lyr_b. Then the time to inform in Lyr_b a neighbor of a vertex of A'_+ or A'_- will basically depend on the number of B-groups that we have to cross. If b is just above a, this number will be small. If b is not far from R, it will be important. In all cases we think it will take a $\Theta(\tau_b)$ steps. Then in $|B|\, D_b$ steps the neighbor of A'_+ or A'_- will inform one of those A'-groups by a push.

One can see that $D_a/|B| = o(\tau_a)$ and $\tau_b = o(\tau_a)$. Thus we can think that the second strategy will be in expectation the faster. We think that there exists a layer $b_\star := b_\star(a)$ such that if $b < b_\star$, this is the case, but if $b > b_\star$, there are too many phases of length τ_b to propagate the rumor into the B-groups, and so the second strategy takes more time.

To conclude, we expect that for small values of b, Lyr_b will help Lyr_a to spread the rumor, while if b is too large, t

4 Conclusion

During this internship, I continued my discovery of the job of researcher. It has been really instructive to work in discrete probabilities, a field that interests me a lot. I appreciated to learn how to deal with typical questions of graph theory.

The study of the push & pull model was a real research problem. I thought that it was difficult given my knowledge. It required reading many articles and running many simulations. I did all simulations myself and I think that it was helpful for the understanding of the propagations. Concerning the theoretical ideas, we are still far from a proof of the expected time of propagation, but Dieter and Marcos will continue to work on it.

Acknoledgement

I would like to sincerely thank Dieter for its availability. He took a lot of time to discuss with me and we often worked together on the theoretical part. It was a pleasure to work with him! Discussions about other topics were also very appreciated. Talking with Marcos was also interesting.

I have to thank all the team of the laboratory for their welcome and their sympathy, especially the probabilistic team (Cédric, Raphaël, Thomas, Karim, Rolland, …). I would also like to mention some PhD students (Giulia, Julie, Laurence, Medhi, Victor, …) and other trainees (David, Dau), I had a good time with them.

References

[BFKL16] T. Bläsius, T. Friedrich, A. Krohmer, and S. Laue. Efficient Embedding of Scale-Free Graphs in the Hyperbolic Plane. In Piotr Sankowski and Christos Zaroliagis, editors, *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:18, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[BFM15] M. Bode, N. Fountoulakis, and T. Müller. On the largest component of a hyperbolic model of complex networks. *The Electronic Journal of Combinatorics*, 22, August 2015.

[BFM16] M. Bode, N. Fountoulakis, and M. Müller. The probability of connectivity in a hyperbolic model of complex networks. *Random Structures & Algorithms*, 49(1):65–94, January 2016.

[BKL15] K. Bringmann, R. Keusch, and J. Lengler. Sampling Geometric Inhomogeneous Random Graphs in Linear Time. *ArXiv e-prints*, November 2015.

[BPK10] M. Boguñá, F. Papadopoulos, and D. Krioukov. Sustaining the Internet with hyperbolic mapping. *Nature Communications*, 1:62, September 2010.

[FK15] T. Friedrich and A. Krohmer. On the diameter of hyperbolic random graphs. *ArXiv e-prints*, December 2015.

[GPP12] L. Gugelmann, K. Panagiotou, and U. Peter. Random Hyperbolic Graphs: Degree Sequence and Clustering. *ArXiv e-prints*, May 2012.

[KM14] M. Kiwi and D. Mitsche. A bound for the diameter of random hyperbolic graphs. *ArXiv e-prints*, August 2014.

[KM17] M. Kiwi and D. Mitsche. On the second largest component of random hyperbolic graphs. *ArXiv e-prints*, December 2017.

[KPK+10] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, September 2010.

[LS17] S. Lamm and C. Schulz. Communication efficient algorithms for generating massive networks. 2017.

[MS17] T. Müller and M. Staps. The diameter of KPKVB random graphs. *ArXiv e-prints*, July 2017.

[Pen17] M. Penschuck. Generating Practical Random Hyperbolic Graphs in Near-Linear Time and with Sub-Linear Memory. In Costas S. Iliopoulos, Solon P. Pissis, Simon J. Puglisi, and Rajeev Raman, editors, *16th International Symposium on Experimental Algorithms (SEA 2017)*, volume 75 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[vÖLM16] M. von Looz, M. Özdayi, S. Laue, and H. Meyerhenke. Generating massive complex networks with hyperbolic geometry faster in practice. *ArXiv e-prints*, June 2016.