The Journal of Supercomputing (2022) 78:15250–15268
https://doi.org/10.1007/s11227-022-04485-5

Hyperbolic trees for efficient routing computation

Zalán Heszberger[1,2,3,4]

Accepted: 24 March 2022 / Published online: 15 April 2022
© The Author(s) 2022

Abstract
Complex system theory is increasingly applied to develop control protocols for distributed computational and networking resources. The paper deals with the important subproblem of finding complex connected structures having excellent navigability properties using limited computational resources. Recently, the two-dimensional hyperbolic space turned out to be an efficient geometry for generative models of complex networks. The networks generated using the hyperbolic metric space share their basic structural properties (like small diameter or scale-free degree distribution) with several real networks. In the paper, a new model is proposed for generating navigation trees for complex networks embedded in the two-dimensional hyperbolic plane. The generative model is not based on known hyperbolic network models: the trees are not inferred from the existing links of any network; they are generated from scratch instead and based purely on the hyperbolic coordinates of nodes. We show that these hyperbolic trees have scale-free degree distributions and are present to a large extent both in synthetic hyperbolic complex networks and real ones (Internet autonomous system topology, US flight network) embedded in the hyperbolic plane. As the main result, we show that routing on the generated hyperbolic trees is optimal in terms of total memory usage of forwarding tables.

Keywords Hyperbolic trees · Routing · Complex networks · Scale-free distribution

✉ Zalán Heszberger
heszi@tmit.bme.hu

[1] Budapest University of Technology and Economics, Budapest, Hungary

[2] Department of Telecommunications and Media Informatics, Budapest, Hungary

[3] High Speed Networks Laboratory, Budapest, Hungary

[4] MTA-BME Information Systems Modeling Research Group, Budapest, Hungary

1 Introduction

With the initial deployment of the third generation of mobile systems, networked IT infrastructures have been becoming increasingly complex with each newer generation. The widespread introduction of computational clouds to support the pervasive AI formed by a large number of intelligent and dumb devices under the framework of IoT (Internet of Things) an unprecedented, highly complex networking environment have been evolved. The IT service ecosystem is further complicated by the software first paradigm shift, that is when software-based configuration control is preferred over changing the underlying hardware. Hyperscaler companies appeared to offer cloud-native scalable computational resources on complex connected data-center infrastructures. To handle the inextricably connected heterogeneous computational environment, recently complex networking solutions have been proposed [1]. An important subproblem in this line of research is obtaining resource efficient routing algorithms in large networks. Short paths are desired for low latency applications or to scenarios where efficient network resource allocation is the primary objective, while low memory usage is favorable, e.g., in small, energy-efficient communication devices like IoT sensors [2]. Memory efficiency is also advantageous to make unstructured complex networking environments scalable for navigation, as in the case of the BGP tables in the Internet interdomain routing system [3].

In the paper, navigational trees are proposed to establish scalable routing of large networks in terms of total memory usage of the forwarding tables employed. The proposed tree, as a kind of "navigation skeleton," is considered to be the set of edges in the network made available for communication. Trees play an essential role in network operation and management. Carefully chosen ones can be considered skeletons of more complex, real networks, acting as a scaffold for certain vital functions like routing [4], navigation [5], cluster analysis or broadcasting [6]. For example, the so-called minimum (weight) spanning tree can efficiently be used for designing routing algorithms and protocols in computer and communication networks. The spanning tree of a network contains all the nodes with only a subset of the edges. That is, it is algorithmically generated from the whole original network. In this paper, we follow a completely different approach. Led by the assumption that the evolution of real networks is toward economical ways of implementing efficient navigability, we propose an essentially hierarchical structure—a hyperbolic tree—as a vital part of complex networks. Only hyperbolic plane coordinates of network nodes are used, and a simple rule is applied to establish tree edges. The rule significantly differs from the ones used for hyperbolic complex network generation [7], and no other structural properties of the network are considered in the hyperbolic tree generation.

The main contributions of the paper are the following:

- First, it is shown, both analytically and numerically, that the degree distribution of the hyperbolic trees generated are scale-free, that is, they follow a power-law function.

- Second, it is also demonstrated that the hyperbolic trees are highly present in synthetic hyperbolic networks, as well as in real networks embedded in the hyperbolic plane.
- Third, as the highlighted contribution of the paper, it is shown based on extensive numerical investigations, that the expected memory usage of hop-by-hop routing with hyperbolic trees has optimal scaling with respect to the size of the trees.

2 Related work

The characterization of real network structures through binding evolutionary forces to properties like being navigable [8], searchable [9] or controllable [10] is a popular approach since the birth of complex network theory. The principal idea behind many of them is the assumption that network formation is led by a hidden metric among the nodes. The idea of characterizing networks by embedding them into the hyperbolic space [11] stems from the observation that artificial generation of nature-like hyperbolic networks can readily be done by surprisingly compact generative models. Related studies [12, 13] show that scattering points as nodes on the hyperbolic plane evenly within a disk and adding edges between nodes being closer than a threshold directly give rise to scale-free networks similar to many real networks. Applications of this line of research consist of, e.g., designing routable data centers [14] or fitting protein interaction networks [15]. Recent developments in the area propose hyperbolic routing techniques to achieve low delay data transmission in NDN (Named Data Networks) [16] or robust learning strategies in Hyperbolic Deep Neural Networks [17]. In the paper, we aim to apply hyperbolic embedding-based network design to make the network routable using small table sizes in the participating nodes. Related research works in the area [11, 12] concentrated on making the network navigable directly by the assigned hyperbolic coordinates. Current paper assumes no structure built into the assigned coordinates. It uses any general popularity versus similarity type hyperbolic embedding schemes [18, 19] and shows that the required memory to successfully route the entire network scales optimally with respect to the number of network nodes. For that purpose, a tree is generated using the coordinates derived from the initial embedding, and routing is restrained to the edges of the tree. Although the tree may contain extra edges not existing in the original network, according to experimental results, the excess link ratio quickly diminishes as network size increases.

3 The hyperbolic plane

The hyperbolic plane (the two-dimensional hyperbolic space) is a metric space, hence there is a well-defined distance calculation such as (hyperbolic cosine theorem)

$$\cosh d(u,v) = \cosh r_u \cosh r_v - \sinh r_u \sinh r_v \cos \phi \tag{1}$$

where r_u and r_v are the radial components of the polar coordinates of points u and v, and $\phi = \phi_u - \phi_v$ is the difference of the angular components of the polar coordinates

[20, 21]. Note that it significantly differs from the Euclidean cosine theorem. The fundamental nature of hyperbolic space is its constant negative curvature, which is not specified here because we do not directly need it. From now on, we assume unit negative curvature, which corresponds to the distance calculation formula above. The direct consequence of negative curvature is the exponential behavior; for example, the circumference and area of circles are exponential functions of the radii instead of polynomials like in the Euclidean space. The area of a hyperbolic disk with radius R is

$$A_R = 2\pi(\cosh(R) - 1)\,. \tag{2}$$

A further exciting and counter-intuitive property of the hyperbolic plane is that the area of triangles is bounded above by π and equals to (in case of unit negative curvature)

$$A_{triangle} = \pi - \alpha - \beta - \gamma \tag{3}$$

where α, β, γ are the angles of the triangle. The immediate consequence is that the triangle areas can usually be neglected, which can significantly simplify the calculations. For example, the area of a circle sector with $R = 5$ and $\phi = \pi/2$ (a quarter circle) is $\frac{\pi}{2}(\cosh 5 - 1) \approx 115$ while the area of the right-angled triangle with sides 5 a is less than $\pi/2 \approx 1.57$. This means that the area of a circle sector is mainly concentrated to the circle segment. The above equations represent the three main properties, the careful use of which provides the basis of our calculations and derivations.

4 Hyperbolic trees

For generating hyperbolic random trees, let the model be the following. Let N points having uniform distribution random coordinates be generated on a hyperbolic disk of radius R. The uniform distribution means that the node density (i.e., the number of nodes per unit area) on the R–disk is constant. From this and (1), it follows that the probability density of the angle coordinates of the points is uniform between 0 and 2π, while the radial coordinate density is given by

$$\rho(r) = \frac{\sinh r}{\cosh R - 1}\,. \tag{4}$$

In a generated point set, the coordinates of the points are randomly sampled from the distributions above. The construction of the hyperbolic tree is given by the following rules. Each point in the set generated corresponds to a node in the tree. First, the points are ordered by increasing radial coordinates. The first node (the point with the smallest radial coordinate) is the root of the tree. Then the i^{th} node ($i = 2, ..., N$) is connected to the closest one among $(1, ..., i-1)$, that is to the closest one having smaller radial coordinate. An example can be seen in Fig. 1 in which $N = 200$ nodes are placed uniformly on a disk of radius $R = 9$. For illustration purposes, on the figure, the native representation of hyperbolic space is used; that is, in the drawings, we use hyperbolic coordinates as if they were Euclidean. Although this may cause

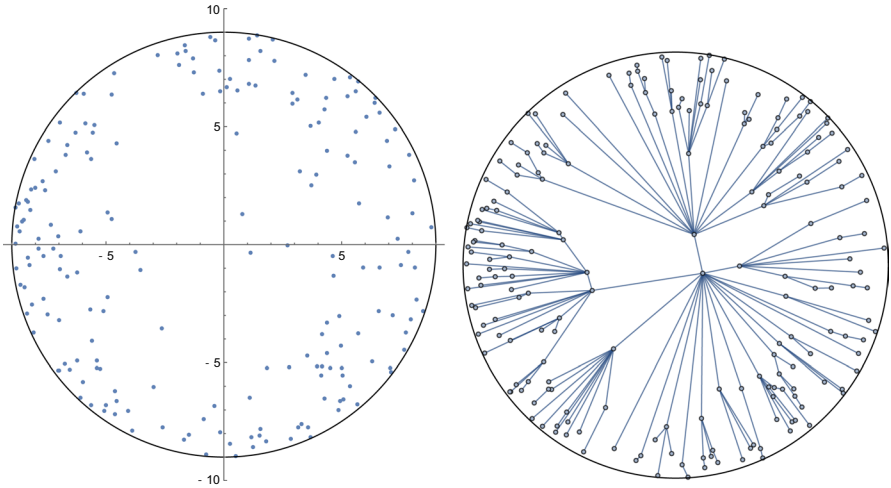

Fig. 1 Points generated on a radius $R = 9$ hyperbolic disk according to uniform distribution (left side) and the hyperbolic tree inferred from this point (right side). One can observe that several links in the tree are oriented toward radial directions due to the explanation of hyperbolic distance characteristics

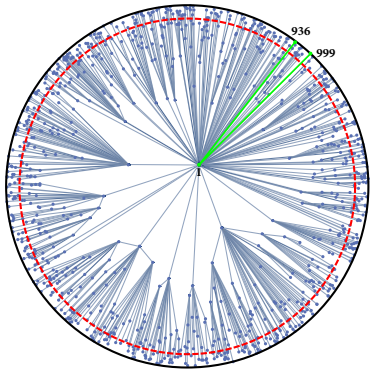

Fig. 2 A hyperbolic tree with 1000 nodes. One can observe the strong hierarchy and the wide spectrum of node degrees. There are exactly half of the nodes (500) inside the red dashed circle. Areas and node densities are not what they seem. Distances also significantly differ from Euclidean one, e.g., node 936 and 999 closer to node 1 than to each other (green dots), see explanation in the text

some strange phenomena in the figures, the textual descriptions of the algorithms and examples become more comfortable to interpret. As the calculations and derivations do not rely on the properties of any specific representations, there is no reason to use more sophisticated models like the Poincarè disk or the Klein model in our investigations.

In Fig. 2, a more complex example can be seen with $R = 14$, $N = 1000$. One can immediately notice that the nodes seem to be non-uniformly distributed on the disk. This is due to the non-isometric nature of the native representation. Note that all other representation models in the Euclidean plan lack isometry, the hyperbolic plane simply "too big", e.g., contains exponentially larger amount of space than the Euclidean one. In the case of the disk with a radius of 14, the total area is $2\pi(\cosh 14 - 1) \approx 3.78 \times 10^6$. The half of this area is 1.8910^6, which can be covered

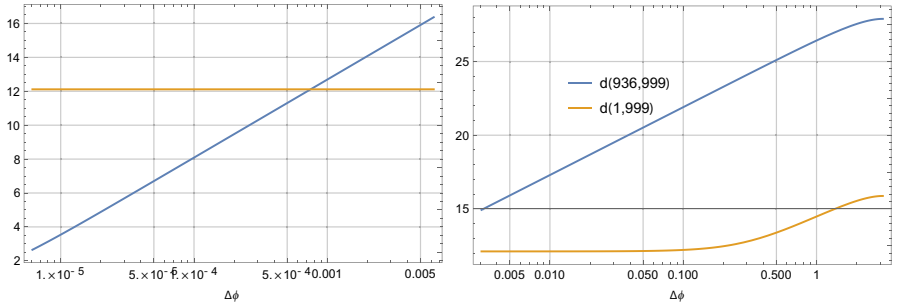

Fig. 3 Hyperbolic distances between nodes 936,999 and nodes 1,999 are presented. In case of similar radial coordinates (r_{936}, r_{999}) the distance is quickly increasing with angular difference, while in case of large radial coordinate difference (r_1, r_{999}) the distance function is quite flat

by a disk with radius 13.3! Due to the uniform node distribution, half of the nodes are expected within the disk of 13.3. In the particular realization in Fig. 2, exactly 500 nodes are inside the 13.14 radius disk (shown by red dashed line in the figure) while the other 500 ones have radial coordinate values within 13.14 and 14. This clearly illustrates the exponential behavior of a hyperbolic plane: areas and node densities are not what they seem in the figures.

Another strange phenomenon in Figs. 1 and 2 is that the orientation of most of the connections lies close to the radial direction. It seems that lots of node pairs (especially in the outer rim of the disk) should have been connected because they look to lie close enough to each other; further evidence that the hyperbolic distance calculation significantly differs from the Euclidean one. We demonstrate that with simple examples. In Fig. 2, the green connections highlight the first node (with the smallest radial coordinate) connected to nodes 936 and 999. The polar coordinates are as follows: $(\phi_1, r_1) = (1.07836, 1.88073)$, $(\phi_{936}, r_{936}) = (0.93668, 13.8972)$, $(\phi_{999}, r_{999}) = (0.829746, 13.9966)$. According to the hyperbolic cosine law (1), the distances are $d(1, 936) = 12.2075$, $d(1, 999) = 12.6142$ $d(936, 999) = 22.0355$. Thus nodes 936 and 999 turn out to be much farther from each other than it seems in the native representation, becoming the first node the closest neighbor to both. Hence, they should not be connected. In fact, if the radial coordinates of two nodes are close to each other, the hyperbolic distance rapidly increases with increasing angular difference, especially in the range of small values. Figure 3 illustrates this behavior, in which the angular coordinate ϕ_{999} of node 999 is varying from ϕ_{936} to $\phi_{936} - \pi$.

One can observe that node 999 would be closer to node 936 only in an extremely small range of $0 < \Delta\phi < 0.00076$; for all larger $\Delta\phi$ $d(936, 999) > d(1, 999)$ holds.

5 Degree distribution of hyperbolic trees

In this section, the degree distribution of the nodes in the hyperbolic tree is investigated. First approximate analytical derivations are performed, than numerical experiments are presented. The derivation of the approximate degree distribution formula consists of three main steps. In the first step, the connection probability $p(u, v)$ for

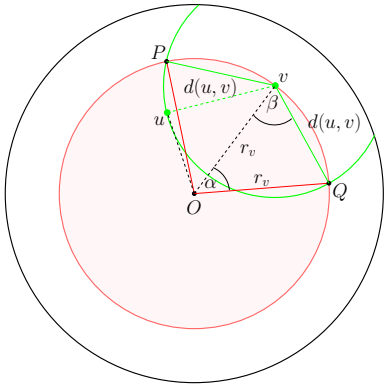

Fig. 4 Geometric illustration for the calculation of the connection probability $p(u, v)$: Node v is connected to node u ($r_u < r_v$) if no nodes are contained in the intersection of the O–centered disk (colored in red) with radius r_v and the v–centered disk (green line) with radius $d(u, v)$

a node pair u, v ($r_u < r_v$) is derived. In the second step, the conditional expected degree $\bar{k}(r_u)$[1] is computed based on $p(u, v)$ and the probability density of node v. In the third step, the degree distribution is inferred from $\bar{k}(r_u)$ with appropriate deconditioning.

Assume that N points are randomly placed on a disk of radius R evenly distributed over its area and a tree is generated applying the previously described algorithm. We can establish the probability $p(u, v)$ that an arbitrary node pair (u, v) is connected. More specifically, $p(u, v)$ is the probability that u and v are connected under the condition that $r_u < r_v$.[2] In Fig. 4, a node pair u, v are shown with $r_u < r_v$. According to the tree generation rule, v is connected to the closest node which has a smaller radial coordinate than r_v. In a geometric interpretation, v connects to the closest node inside the O-centered disk of radius r_v (red disk in the figure). The closest node will be exactly u when no points lie inside the intersection of the O-centered disk with radius r_v and the v-centered disk with radius $d(u, v)$ (green disk in the figure). The probability that all the $N-2$ points fall outside the intersection area is $1 - \frac{A_{QP}}{A_R}$, where A_{QP} denotes the area of the intersection, and $A_R = 2\pi(\cosh R - 1)$ is the area of the whole disk. It follows that

$$p(u, v) = \left(1 - \frac{A_{QP}}{A_R}\right)^{N-2} = \left(1 - \frac{A_{QP}}{A_R}\right)^{A_R \frac{N-2}{A_R}} \approx \mathrm{e}^{-\delta A_{QP}}, \tag{5}$$

where δ is the node density: $\delta = \frac{N}{A_R} \approx \frac{N-2}{A_R}$ What remains to be done to calculate the probability $p(u, v)$ is to establish A_{QP}. One can observe in Fig. 4 that A_{QP} is equal to the sum of the two circle sectors POQ (red, sector angle is 2α) and PvQ (green, sector angle is 2β) minus the area of the two triangles PQO and PQv:

[1] We introduce the short notation $\bar{k}(r_u)$ for $\mathrm{E}[k(u)|r_u]$, that is, the conditional expectation of the degree of a node u, whose radial coordinate is r_u.

[2] The notation $p(\text{u connectedtov}|r_u, r_v, r_u < r_v)$ could be more appropriate, but for brevity we use $p(u, v)$.

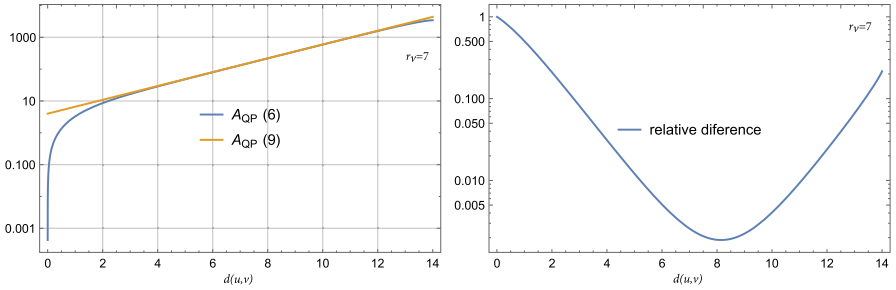

Fig. 5 Different approximations of A_{QP} and their relative difference

$$A_{QP} \approx 2\alpha(\cosh r_v - 1) + 2\beta(\cosh d(u,v) - 1), \tag{6}$$

where the area of the two triangles $2(\pi - \alpha - 2\beta)$ is already neglected. The angles α and β can be determined by applying the hyperbolic cosine law (1) to the triangle vOQ with edges OQ and vQ. From these

$$\alpha(d_{uv}, r_v) = \arccos \frac{\cosh^2 r_v - \cosh d_{uv}}{\sinh^2 r_v} \tag{7}$$

and

$$\beta(d_{uv}, r_v) = \arccos \frac{\cosh r_v(\cosh d_{uv} - 1)}{\sinh r_v \sinh d_{uv}} \tag{8}$$

Now the Eqs. (6), (7) and (8) can be combined to get A_{QP} as a function of $d(u, v)$ and r_v. Because the formula is a bit lengthy and not very expressive, it is not repeated here. However, based on a sequence of approximations a much simpler and useful expression can be obtained as

$$A_{QP} \approx 4\mathrm{e}^{\frac{d(u,v)}{2}} \tag{9}$$

by which we get a simple formulae for the approximation of the connection probability:

$$p(u,v) \approx \mathrm{e}^{-\delta 4\mathrm{e}^{\frac{d(u,v)}{2}}}. \tag{10}$$

At this point, it is practical to postpone the in-depth analysis of the details and quality of these approximations, as two more weighing functions are to appear in the computation of $\bar{k}(r_u)$. Instead, an illustrative example is shown in Fig. 5 to study the performance of the different approximations of A_{QP} , where $r_v = 7$ is fixed and $d(u, v)$ is running from 1 to 14 (note that $d(u, v) < 2r_v$ holds for all (u, v), $r_u < r_v$).

One can observe that for a wide range of $d(u, v)$ values, the simpler approximation (9) is quite close to the more tedious one; the relative difference is below 5% when $3 < d(u, v) < 13$. Because the distance $d(u, v)$ is also a function of r_u and r_v itself, it is worth expanding it exploiting (1) as

$$\cosh d(u,v) = \cosh r_u \cosh r_v - \sinh r_u \sinh r_v \cos\phi, \tag{11}$$

where ϕ is the difference between the angle coordinates of u and v. Using the approximations $\cosh x \approx \frac{e^x}{2}$ and $\sinh x \approx \frac{e^x}{2}$ when x is not too small, one can write

$$\frac{e^{d(u,v)}}{2} = \frac{e^{r_u+r_v}}{4}(1-\cos\phi)\,. \tag{12}$$

Based on this equation, we get

$$e^{\frac{d(u,v)}{2}} \approx e^{\frac{r_u+r_v}{2}} \sqrt{\frac{1-\cos\phi}{2}}. \tag{13}$$

Using the remarkable identity $\sqrt{\frac{1-\cos\phi}{2}} = \sin\frac{\phi}{2}$ a further, more useful formula can be obtained for $p(u, v)$ for subsequent derivation as

$$p(u,v) \approx e^{-4\delta e^{\frac{r_u+r_v}{2}} \sin\frac{\phi}{2}}. \tag{14}$$

Now we can continue with the derivation of $\bar{k}(r_u)$. Roughly speaking, $\bar{k}(r_u)$ is the expected number of neighbors of u when all nodes v, $(r_u < r_v < R)$ are counted for which u is the closest. There is surely one more "downward" connection of u to a node with an even lower radial coordinate due to the tree generation rule (except when u has the smallest r_u, but this exception does not influence the results). More formally, $p(u, v)$ is to be integrated with the probability densities of r_v and ϕ $(\phi) := \phi_v - \phi_u$ (and a constant 1 should be added), that is

$$\bar{k}(r_u) = 1 + N \int_{r_v=r_u}^{R} \int_{\phi=0}^{2\pi} p(u,v) \frac{1}{2\pi} \frac{\sinh r_v}{\cosh R - 1} d\phi dr_v\,. \tag{15}$$

First, consider the integral with respect to the angle difference ϕ:

$$\int_{\phi=0}^{2\pi} p(u,v) d\phi\,. \tag{16}$$

Interestingly enough, if we use approximation (14), it can be expressed exactly with two distinguished functions: the zero-order modified Bessel function $\boldsymbol{I}_0(x)$ and the zero-order modified Struve function $\boldsymbol{L}_0(x)$

$$\int_{\phi=0}^{2\pi} e^{-x\sin\frac{\phi}{2}} d\phi = 2\pi\big(\boldsymbol{I}_0(x) - \boldsymbol{L}_0(x)\big), \tag{17}$$

where $x = 4\delta e^{\frac{r_u+r_v}{2}}$. It is known that for large x, $\boldsymbol{I}_0(x) - \boldsymbol{L}_0(x)$ quickly tends[3]. to $\frac{2}{\pi x}$, therefore the right-hand side of the equation above is approximated by

[3] The asymptotic series expansion of $\boldsymbol{I}_0(x) - \boldsymbol{L}_0(x)$ at $x = \infty$ starts with $\frac{2}{\pi x}$, see [22] . The asymptotic series expansion in this case means that $\lim_{x\to\infty} \frac{\boldsymbol{I}_0(x)-\boldsymbol{L}_0(x)}{\frac{2}{\pi x}} = 1$

$$\frac{4}{x} = \frac{1}{\delta} e^{-\frac{r_u + r_v}{2}} . \tag{18}$$

Putting it back to Eq. (15), we get

$$1 + N \int_{r_v = r_u}^{R} \frac{1}{\delta} e^{-\frac{r_u + r_v}{2}} \frac{1}{2\pi} \frac{\sinh r_v}{\cosh R - 1} dr_v . \tag{19}$$

Using the definition of node density $\delta = \frac{N}{2\pi(\cosh R - 1)}$, we get

$$1 + \int_{r_v = r_u}^{R} e^{-\frac{r_u + r_v}{2}} \sinh r_v dr_v \approx 1 - 1 + e^{\frac{R - r_u}{2}} . \tag{20}$$

Finally, we get the stunningly simple formula

$$\bar{k}(r_u) \approx e^{\frac{R - r_u}{2}} . \tag{21}$$

The final formula reflects the property observed in the previous numerical examples, namely that nodes with smaller radial coordinates tend to have a much higher number of connections. There is an interesting and easy way to perform a sanity check on the formula. All trees with N nodes have exactly $N - 1$ links, that is the grand average node degree is $\bar{k} = 2(N - 1)/N \approx 2$. This means that the result of integrating $\bar{k}(r_u)$ with respect to the density r_u should be very close to 2:

$$\bar{k} = \int_{r_u = 0}^{R} e^{\frac{R - r_u}{2}} e^{r_u - R} dr_u = 2(1 - e^{-\frac{R}{2}}), \tag{22}$$

which differs from 2 with a negligible factor when R is not too small (for example, in case of $R = 10$ $\bar{k} = 1.98652$).

As a third step, the complement cumulant degree distribution is derived, i.e., the probability that an arbitrary node degree is larger than a given value k, $P(degree > k)$ is computed. Here we present an approximate and more perceptible reasoning instead of a sophisticated but tedious one. If we accept the approximation (21), than from its monotonicity if follows, that nodes with higher *expected* degrees than a given k are *exactly* inside the circle of radius $r_u(k)$, where $r_u(k)$ is the inverse function of (21). One can also state that nodes with degrees *exactly* higher than a given k are *expectedly* inside the circle with radius $r_u(k)$. These two statements are approximately equivalent; therefore, $P(degree > k)$ can be approximated by the ratio of the area of $r_u(k)$-disk and the R-disk

$$P(degree > k) \approx \frac{2\pi(\cosh r_u(k) - 1)}{2\pi(\cosh R - 1)} \approx \frac{e^{r_u(k)}}{e^R} . \tag{23}$$

Substituting the inverse function $r_u(k) = R - 2 \log k$ into this equation

$$P(degree > k) \approx k^{-2} . \tag{24}$$

is obtained, that is approximately a power-law distribution with parameter 2.

6 Numerical analysis

In this section, numerical results are presented in connection with synthetic and real networks embedded on the hyperbolic plane. First, the generative model for synthesizing complex networks on the hyperbolic plane is considered [12] . In the simplest case, the network generation rule is as follows: distribute N nodes uniformly on a hyperbolic disk with radius R then connect every node pair which are closer than R. It is analytically shown and numerically confirmed that the degree distribution of such networks follows power law, and the ccdf is proportional to k^{-2} [12, 23].

Two numerical examples are presented for $N = 1000$, $R = 13$ and for $N = 3000$, $R = 14$. In Figs. 6 and 7 the hyperbolic trees generated and the synthetic networks

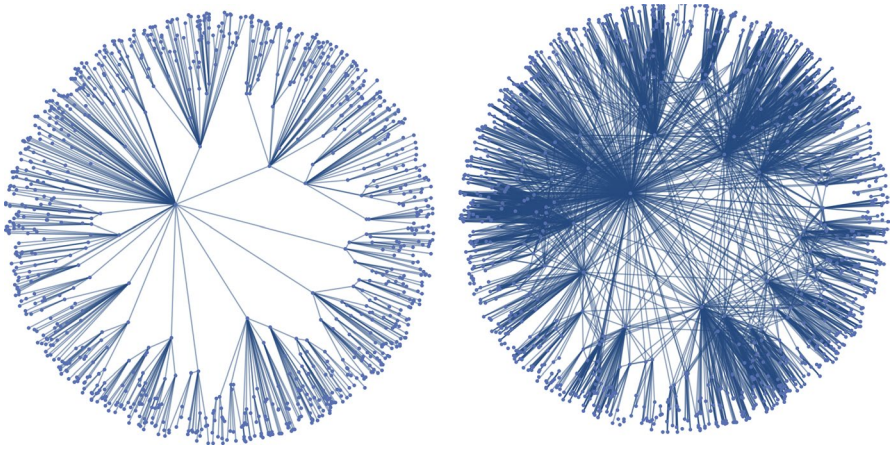

Fig. 6 Hyperbolic tree and a synthetic complex network with $N = 1000$, $R = 13$ for the same set of nodes

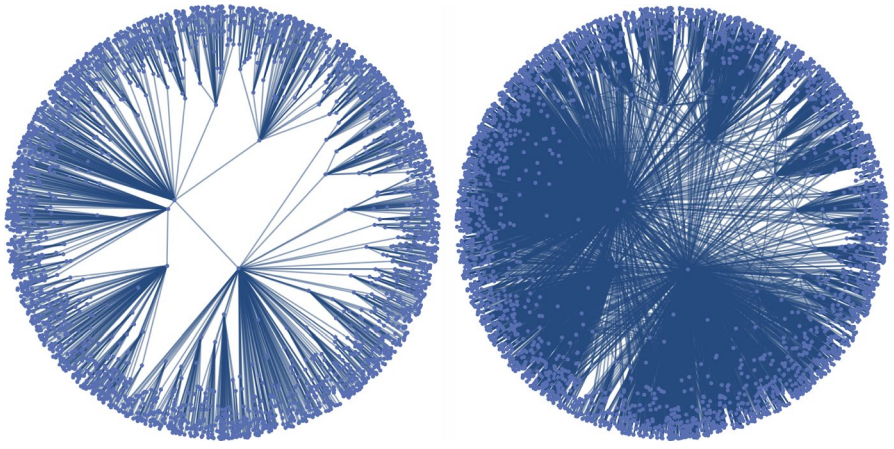

Fig. 7 Hyperbolic tree and a synthetic complex network with $N = 3000$, $R = 14$ for the same set of nodes

can be seen. It is observable that despite the completely different generation rule the structure of the synthetic network is strongly hierarchical and resembles to the tree network. Moreover, 860 of the total 999 links can be found in the original network too. In the second example ($N = 3000$) 2857 of the total 2999 tree links are also present in the synthetic network. The tree inclusion ratio in the first example is $860/999 = 0.86$, while in the second example is as high as $2857/2999 = 0.95$. We have performed more detailed numerical investigations on the inclusion ratio versus network size. The general observation is that the tree inclusion ratio is quickly increasing with increasing size of the network N. Table 1 summarizes the tree inclusion ratios for different sizes of networks averaged over 100 networks for every size.

The structural similarity is confirmed by the histograms for the degree distributions which are shown in Fig. 8. Note that the theoretical function k^{-2} well fits to the measured degree distribution of the tree and is in accordance with the similar decay in the synthetic network for larger values of k.

The degree distributions are also investigated on several networks with different sizes N. In all cases, the decay of the degree distributions is in accordance with the theoretical prediction, especially for larger networks and larger values of degree. In Fig. 9 the histograms of the degrees are shown for $N = 1000, 2000, 5000, 10000$. One can observe that for small values of degrees ($k = 1..5$), the theoretical function underestimates the measured frequency in all sizes a bit, and the estimation becomes more accurate for $N > 2000$ and $k > 5$. Note that the very tail of the histograms are statistically less reliable due to the lack of enough data in the vicinity of the maximum degree, in these regions (i.e., $N = 10000$, $k = 60$–100) well fit to the theoretical curve cannot be expected.

Table 1 Tree inclusion ratios for different size of networks

Number of nodes N	1000	3000	5000	7000	10000
Tree inclusion ratio	0.863	0.948	0.972	0.987	0.992

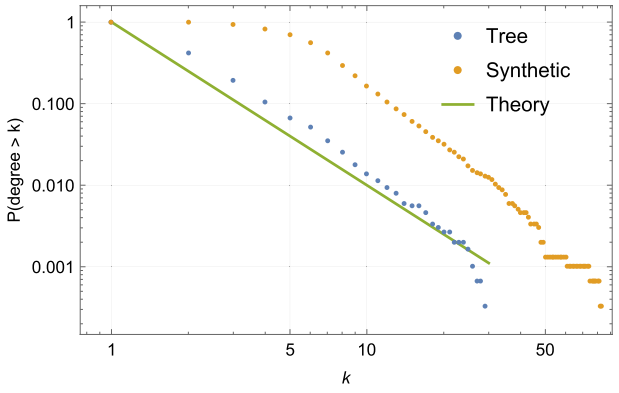

Fig. 8 Degree distributions of a synthetic complex network and the corresponding hyperbolic tree

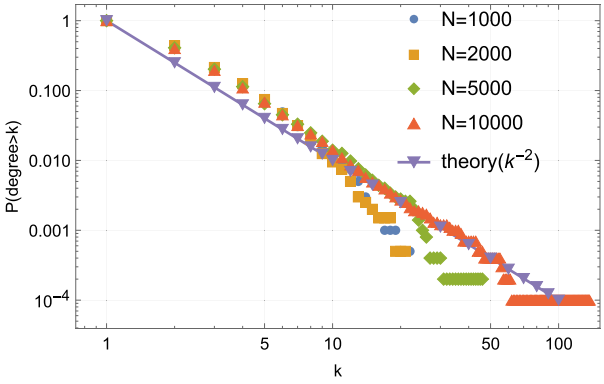

Fig. 9 Empirical degree distributions (histograms) for four different hyperbolic tree sizes

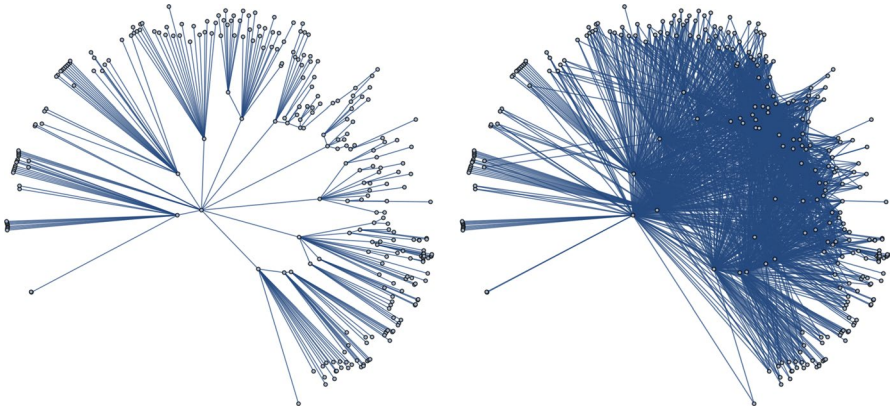

Fig. 10 The US flight network (right) embedded into the hyperbolic plane, and the corresponding hyperbolic tree (left)

The hyperbolic tree networks are also generated and tested on two types of real networks embedded in the hyperbolic plane. Here we do not outline the embedding process; just note that the embedding process does not use algorithms based on any trees inferred from the original network. For more details of the embedding see [12, 24, 25]. Thus, our hyperbolic tree generation method does not "know" directly anything about any trees contained in the original real networks.

The first network is the US flight network, that was downloaded from the Bureau of Transportation Statistics http://transtats.bts.gov/ on 5 November 2017. It consists of 283 nodes and 1973 edges. In the network, the nodes are US airports. Two nodes are linked if a direct flight connects them. The hyperbolic tree and the original network are drawn in Fig. 10. The hyperbolic tree has 282 links of which 246 are also in the original network. It corresponds to a 87.2% tree inclusion ratio. The degree distributions of the hyperbolic tree and the original flight network can be seen in Fig. 11. One can observe that the decay of the histograms is different; however, both

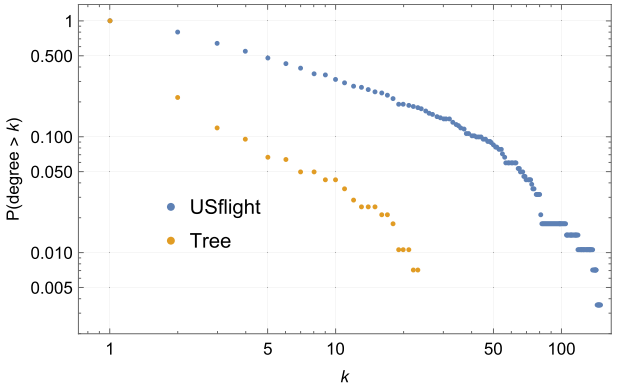

Fig. 11 US flight network and its tree degree distribution. The tree is generated based only on the hyperbolic coordinates

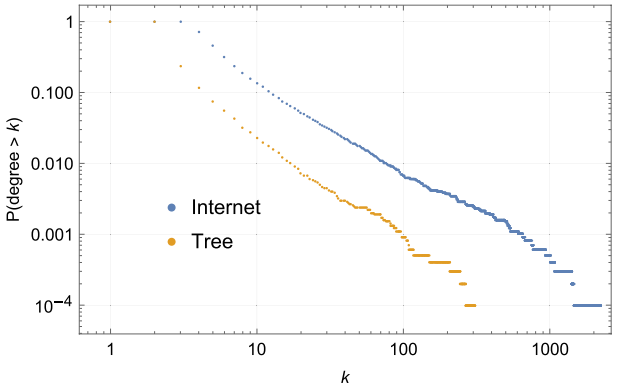

Fig. 12 Internet AS level topology (inner core of 10000 nodes) and its tree degree distribution

are power-law functions which confirms our theoretical results. The flight network is embedded into a 11.3 radius disk determined by the embedding algorithm, so the average node density on this disk is about 0.0035.

The other network is the much larger Internet AS level topology. The corresponding data set representing the global Internet structure at the autonomous system (AS) level is from [26]. In this network, we use the inner core of 10000 nodes (out of the total 23748), the number of edges is 40605, and the network is embedded into a 26.87 radius disk. In this case, the node density is much lower, 6.83×10^{-9}. The degree distribution can be seen in Fig. 12. Similar power-law behavior also appears here, the difference between the decay rates is smaller than in the previous case. The inclusion ratio of the hyperbolic tree in the original network is also as high as 84.7%. In both cases, the difference between the exponents of the power functions can be attributed to the fact that real network embeddings usually cause non-uniform node distributions. According to our numerical experiments,

this non-uniformity results in a smaller change in the tree degree distribution than in that of the original network.

7 Routing efficiency of hyperbolic trees

In this section, the memory requirements of hop-by-hop routing is analyzed numerically when hyperbolic tree is used as a scaffold for the routing function. The analysis also contains the comparison to other spanning-tree-based routing algorithms. As presented in [4], the calculation of the information theoretical entropy of routing tables turned out to be a useful tool for estimating the total memory requirements. The hop-by-hop routing forwarding table of a node in an N-node network is an $N-$length string with the alphabet consisting of the node neighbor identifiers. The i^{th} element of the table is j at node k means that the packet arrived at node k should be forwarded to node j (as next hop) in order to reach node i. The entropy of the node k routing table can be defined as

$$H_k = \sum_{i=1}^{\delta_k} \frac{n_i}{N} \log \frac{N}{n_i}\,, \tag{25}$$

where n_i is the number of entries with the i^{th} neighbor identifier, and δ_k is the number of neighbors of node k. NH_k can be interpreted as the information theoretic lower bound of the required number of bits to encode the whole routing table, and therefore, the required number of bits to represent all routing tables is at least $N\sum_{k=1}^{N} H_k$. Therefore, from scalability point of view, the dependency of the whole routing entropy of the network $H(N) := \sum_{k=1}^{N} H_k$ with respect to N has an utmost importance.

For the numerical analysis, first, a sequence of hyperbolic trees was generated with different sizes ($N = 100, 200, 500, 1000, 2000, 5000, 10000$) ranging two orders of magnitudes of N. Then the forwarding tables of the hop-by-hop routing were inferred, and the entropy of the tables was calculated. At all sizes of N 100, random trees were generated in order to eliminate statistical fluctuations. The following table shows the average entropies $\bar{H}(N)$ (standard deviations in brackets) of hyperbolic trees with different sizes (Table 2).

One can observe from the data set that the average total entropy $\bar{H}(N)$ increases with increasing N in a sub-linear manner. More accurate trend can be identified, if the data is plotted with a logarithmic scale x axis and a linear scale y axis.

As can be seen in Fig. 13 the average entropy values well fit on a line in this log-linear scaling, which means that the increase of $\bar{H}(N)$ with respect to N might approximately be in the order of $O(\log N)$. This is a very nice

Table 2 Average entropies of hyperbolic trees over different sizes

N	100	200	500	1000	2000	5000	10000
$\bar{H}(N)$	16.771	18.872	21.635	23.719	25.801	28.534	30.657

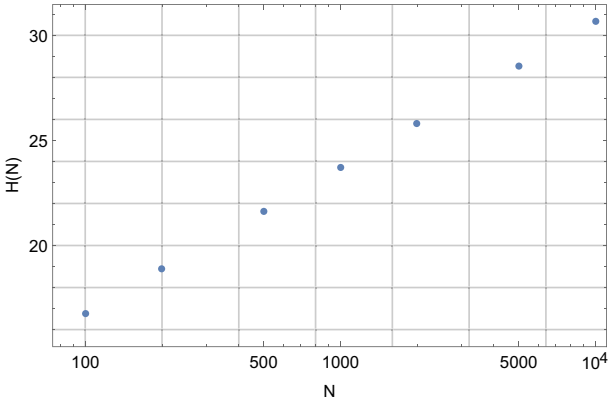

Fig. 13 Average entropy in the function of N

Table 3 Entropy-based lower bounds and actual values (by Huffman coding) for the memory usage of routing tables (in bits)

N	100	200	500	1000	2000	5000	10000
$N\bar{H}(N)$	1677	3774	10817	23719	51600	142670	306570
Huffman codes	1717	3851	11102	24330	52629	144950	313570

scaling property because the total amount of memory (measured in bits) $N\bar{H}(N)$ required to store all routing tables in an $N-$size hyperbolic tree is in the order of $O(N \log N)$ which is the best possible scaling achievable in case of trees [4]. The approximate linear behavior of $H(N)$ in the log-linear scaling means that it is worth fitting a linear model to the pairs $(\log N, H(N))$, which, in this case, is $H(N) = 2.917 + 3.010 \log N$. We have also tested the validity (the prediction power) of this linear fit for such values of $(\log N, H(N))$ which were not involved in the linear model generation. For example, the total entropy (averaged over 100 hyperbolic trees generated) in case of $N = 4000$ and $N = 6000$ are $H(4000) = 27.840$ and $H(6000) = 29.097$, respectively. The linear model predicts 27.882 and 29.103 for these entropies, the relative error of the prediction in these cases (and in many other examples, too) is far below 1%.

Because $N\bar{H}(N)$ is a lower bound for the expected total memory requirement of routing, we have also implemented the binary codes of the routing tables for all sizes mentioned above and counted the actual total memory usage of the forwarding tables. Huffman coding has been used, which is suitable for encoding routing tables due to its prefix nature. The actual memory usage is very close to the entropy-based lower bound in all cases (the relative error is below 3%), Table 3 shows this observation.

In our next research task, we are going to show analytically this scaling behavior of routing table entropy in hyperbolic random trees.

8 Discussion

The scalability of memory usage in arbitrary trees can widely vary [4] depending on the structure of the tree. For example, in case of pure binary trees the total routing entropy function scales as $O(\log^2(N))$, where N is the number of nodes in the tree. Another extreme example is the star network in which the total entropy is exactly

$$H_{star}(N) = \frac{2N-1}{N}\log(N) = O(\log(N)) \,. \tag{26}$$

The above scaling is desirable, but in case of general networks, it is impossible to generate spanning trees out of the original network edges that have similar structural characteristics. Furthermore, standard spanning tree generation algorithms neither can provide the structural similarities to the original network, which might be a further reason of the diverse scaling behavior of tree routing in general. Reversing the train of thoughts, one can suspect that increasing trees showing structural similarities might result in a focused, and well-characterizable scaling behavior. Because the hyperbolic plane can be used to generate increasing sequence of networks sharing similar structural properties, we expected the same in our hyperbolic tree generation method. This is confirmed by the observed similar scale-free degree distributions for two orders of magnitude in tree sizes (Fig. 9).

With respect to the high inclusion ratios of hyperbolic trees in real-world complex networks the phenomenon can mainly be attributed to the strong hierarchical and self-similar characteristic of complex networks [27]. Finally, we note that according to our preliminary numerical studies, in the presented hyperbolic tree model, not only the node degrees are distributed in a scale-free manner, but the size of the sub-trees originating from the nodes at different levels has also scale-free distribution. Sub-tree sizes play a direct role in the routing entropy calculations; the good scaling behavior of routing entropy in our hyperbolic trees is likely due to the widely-spread sub-tree size distribution.

9 Conclusion

A new hyperbolic tree generation method has been presented. The generating algorithm is based on the presumption that the successful navigability of many real networks is key to their evolution. Both analytically and by numerical experiments on synthetic and real data, it is shown that the degree distribution of the trees generated is approximately a power function. It has also been demonstrated that the trees are present to a large extent in real and synthetic networks and that hop-by-hop routing with hyperbolic trees is efficient in terms of total memory usage of forwarding tables.

The generation of the synthetic networks is carried out purely in a technical computing environment, as only topology information is to be used for the research. Due to the compact models applied, the complexity of the generation did not exceeded the square of the number of the nodes in the network: The

hyperbolic networks used simple random node generation on a planar disk with edges added depending on the distances between the nodes. Further, routing tables in synthetic, as well as, in networks obtained from public measurement databases are calculated on-the-fly during the edge creation phase. In a real environment, simple modification of spanning tree protocols like STP (Spanning Tree Protocol) are available to carry out that function.

With respect to direct application of the results, the question arises as to how to circumvent the (although rare) problem of the addition of non-existent physical connections in real technological networks. On the other hand, thinking in terms of the softwarized networking paradigm like the SDN (Software Defined Networking), in overlay networks, topology changes like the generation of new edges boils down to a simple reconfiguration process. Further real implementation problems may arise in heavy dynamic environments where the generation time of the navigation tree has strong real-time constraints. In such cases, e.g., auxiliary routing mechanisms may temporarily be applied to eliminate connection service disruption in the system. Finally, our results open further research questions on the deeper relation between navigable hyperbolic trees and the structural evolution of networks.

Funding Open access funding provided by Budapest University of Technology and Economics. Project no. 124171, 128062 and 135606 have been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the K_17, K_18 and ANN_20 funding schemes respectively.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Charalambos S, Marios L, Pavlos A, Christos L, Andreas P (2020) Complex systems: a communication networks perspective towards 6g. IEEE Access, pp 1–1, 05
2. Umar FQM, Xingfu W, Ammar H, Asad K, Adeel A, Teju WF (2020) Torp: load balanced reliable opportunistic routing for asynchronous wireless sensor networks. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp 1384–1389
3. Andre B, Claffy KC (2001) Analysis of routeviews bgp data: policy atoms
4. Attila K, András G, Zalán H, József B, Rétvári G (2020) Tight bounds and optimal address spaces. IEEE/ACM Transactions on Networking, on the Memory Requirement of Hop-by-Hop Routing
5. Gulyás A, Bíró JJ, Kőrösi A, Rétvári G, Krioukov D (2015) Navigable networks as nash equilibria of navigation games. Nat Commun 6:7651
6. Tapolcai J, Bíró J, Babarczi P, Gulyás A, Heszberger Z, Trossen D (2014) Optimal false-positive-free bloom filter design for scalable multicast forwarding. IEEE/ACM Trans Netw 23(6):1832–1845

7. Luca G, Konstantinos P, Ueli P (2012) Random hyperbolic graphs: degree sequence and clustering. In: International Colloquium on Automata, Languages, and Programming, pp 573–585. Springer
8. Kleinberg JM (2000) Navigation in a small world. Nature 406(6798):845–845
9. Watts DJ, Dodds PS, Newman MEJ (2002) Identity and search in social networks. Science 296(5571):1302
10. Guoqi L, Lei D, Gaoxi X, Pei T, Changyun W, Wuhua H, Jing P, Luping S, Eugene SH (2018) Enabling controlling complex networks with local topological information
11. Kleinberg R (2007) Geographic routing using hyperbolic space. In: Proc of INFOCOM
12. Krioukov D et al (2010) Hyperbolic geometry of complex networks. Phys Rev E 82(3):036106
13. Moritz VL, Mustafa SO, Sören L, Henning M (2016) Generating Massive Complex Networks with Hyperbolic Geometry Faster in Practice. In: 2016 IEEE High Performance Extreme Computing Conference (HPEC), pp 1–6
14. Márton C, András G, Attila K, Balázs S, Gergely B (2012) Poincaré: A Hyperbolic Data Center Architecture. In: 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW), pp 8–16, 06
15. Higham DJ, Rašajski M, Pržulj N (2008) Fitting a geometric graph to a protein–protein interaction network. Bioinformatics 24(8):1093–1099
16. Vince L, Ashlesh G, Beichuan Z, Lixia Z, Rodrigo A, Dmitri K, Lan W (2016) An experimental investigation of hyperbolic routing with a smart forwarding plane in ndn. In: 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), pp 1–10
17. Wei P, Tuomas V, Abdelrahman M, Henglin S, Guoying Z (2021) Hyperbolic deep neural networks: a survey. arXiv preprint arXiv:2101.04562
18. Papadopoulos F, Kitsak M, Serrano M, Boguná M, Krioukov D (2012) Popularity versus similarity in growing networks. Nature 489(7417):537–540
19. Muscoloni A, Thomas JM, Ciucci S, Bianconi G, Cannistraci CV (2017) Machine learning meets complex networks via coalescent embedding in the hyperbolic space. Nat Commun 8(1):1–19
20. Bolyai J (1987) APPENDIX - The Theory of Space (with Introduction, Comments and Addenda by F. Karteszi). North-Holland
21. Anderson JW (2006) Hyperbolic geometry. Springer, New York
22. Abramovitz M, Stegun I (1965) Handbook of Mathematical Functions. Courier Dover Publication, New York
23. Papadopoulos F, Krioukov D, Bogua M, Vahdat A (2010) Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In: Proc. of IEEE Infocom, pp 1–9. IEEE
24. Boguna M, Krioukov D, Claffy KC (2009) Navigability of complex networks. Nat Phys 5(1):74–80
25. Voitalov I, Aldecoa R, Wang L, Krioukov D (2017) Geohyperbolic routing and addressing schemes. ACM SIGCOMM Comput Commun Rev 47(3):11–18
26. Boguna M, Papadopoulos F, Krioukov D (2010) Sustaining the internet with hyperbolic mapping. Nat Commun 1(6):1–8
27. Corominas-Murtra B, Goñi J, Solé RV, Rodríguez-Caso C (2013) On the origins of hierarchy in complex networks. Proc Natl Acad Sci 110(33):13316–13321

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.