Fast approximation of eccentricities and distances in hyperbolic graphs
Victor Chepoi, Feodor F. Dragan, Michel Habib, Yann Vaxès, Hend Alrasheed

To cite this version:

HAL Id: hal-02268468
https://hal.science/hal-02268468
Submitted on 20 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Fast approximation of centrality and distances in hyperbolic graphs

Victor Chepoi¹, Feodor F. Dragan², Michel Habib³, Yann Vaxès¹, and Hend Al-Rasheed²

¹ Laboratoire d’Informatique et Systèmes, Aix-Marseille Univ, CNRS, and Univ. de Toulon
Faculté des Sciences de Luminy, F-13288 Marseille Cedex 9, France
{victor.chepoi, yann.vaxes}@lif.univ-mrs.fr
² Algorithmic Research Laboratory, Department of Computer Science, Kent State University, Kent, Ohio, USA
dragan@cs.kent.edu, halrashe@kent.edu
³ Institut de Recherche en Informatique Fondamentale, University Paris Diderot - Paris7, F-75205 Paris Cedex 13, France
habib@liafa.univ-paris-diderot.fr

Abstract. We show that the eccentricities (and thus the centrality indices) of all vertices of a δ-hyperbolic graph $G = (V, E)$ can be computed in linear time with an additive one-sided error of at most $c\delta$, i.e., after a linear time preprocessing, for every vertex $v$ of $G$ one can compute in $O(1)$ time an estimate $\hat{e}(v)$ of its eccentricity $ecc_G(v)$ such that $ecc_G(v) \leq \hat{e}(v) \leq ecc_G(v) + c\delta$ for a small constant $c$. We prove that every δ-hyperbolic graph $G$ has a shortest path tree, constructible in linear time, such that for every vertex $v$ of $G$, $ecc_G(v) \leq ecc_T(v) \leq ecc_G(v) + c\delta$. These results are based on an interesting monotonicity property of the eccentricity function of hyperbolic graphs: the closer a vertex is to the center of $G$, the smaller its eccentricity is. We also show that the distance matrix of $G$ with an additive one-sided error of at most $c'\delta$ can be computed in $O(|V|^2 \log^2 |V|)$ time, where $c' < c$ is a small constant. Recent empirical studies show that many real-world graphs (including Internet application networks, web networks, collaboration networks, social networks, biological networks, and others) have small hyperbolicity. So, we analyze the performance of our algorithms for approximating centrality and distance matrix on a number of real-world networks. Our experimental results show that the obtained estimates are even better than the theoretical bounds.

1 Introduction

The diameter $diam(G)$ and the radius $rad(G)$ of a graph $G = (V, E)$ are two fundamental metric parameters that have many important practical applications in real world networks. The problem of finding the center $C(G)$ of a graph $G$ is often studied as a facility location problem for networks where one needs to select a single vertex to place a facility so that the maximum distance from any demand vertex in the network is minimized. In the analysis of social networks (e.g., citation networks or recommendation networks), biological systems (e.g., protein interaction networks), computer networks (e.g., the Internet or peer-to-peer networks), transportation networks (e.g., public transportation or road networks), etc., the eccentricity $ecc(v)$ of a vertex $v$ is used to measure the importance of $v$ in the network: the centrality index of $v$ is defined as $\frac{1}{ecc(v)}$.

Being able to compute efficiently the diameter, center, radius, and vertex centralities of a given graph has become an increasingly important problem in the analysis of large networks. The algorithmic complexity of the diameter and radius problems is very well-studied. For some special classes of graphs there are efficient algorithms [3, 16, 25, 30, 32, 33, 42, 54, 60, 62, 79]. However, for general graphs, the only known algorithms computing the diameter and the radius exactly compute the distance between every pair of vertices in the graph, thus solving the all-pairs shortest paths problem (APSP) and hence computing all eccentricities. In view of recent negative results [32, 53], this seems to be the best what one can do since even for graphs with $m = O(n)$ (where $m$ is the number of edges and $n$ is the number of vertices) the existence of a subquadratic time (that is, $O(n^{2-\epsilon})$ time for some $\epsilon > 0$) algorithm for the diameter or the radius problem will refute the well known Strong Exponential Time Hypothesis (SETH). Furthermore, recent work [9] shows that if the radius of a possibly dense graph ($m = O(n^2)$) can be computed in subcubic time ($O(n^{3-\epsilon})$ for some $\epsilon > 0$),
then APSP also admits a subcubic algorithm. Such an algorithm for APSP has long eluded researchers, and it is often conjectured that it does not exist (see, e.g., [84,90]).

Motivated by these negative results, research started devoting more attention to development of fast approximation algorithms. In the analysis of large-scale networks, for fast estimations of diameter, center, radius, and centrality indices, linear or almost linear time algorithms are desirable. One hopes also for the all-pairs shortest paths problem to have \(o(mn)\) time small-constant-factor approximation algorithms. In general graphs, both diameter and radius can be 2-approximated by a simple linear time algorithm which picks any node and reports its eccentricity. A 3/2-approximation algorithm for the diameter and the radius which runs in \(O(mn^{2/3})\) time was recently obtained in [31] (see also [12] for an earlier \(\tilde{O}(n^2 + m\sqrt{n})\) time algorithm and [53] for a randomized \(\tilde{O}(m\sqrt{n})\) time algorithm). For the sparse graphs, this is an \(o(n^2)\) time approximation algorithm. Furthermore, under plausible assumptions, no \(O(n^{2-\varepsilon})\) time algorithm can exist that \((3/2 - \varepsilon)\)-approximates (for \(\varepsilon, \varepsilon' > 0\)) the diameter [83] and the radius [8] in sparse graphs.

Similar results are known also for all eccentricities: a 5/3-approximation to the eccentricities of all vertices can be computed in \(O(m^{3/2})\) time [31] and, under plausible assumptions, no \(O(n^{2-\varepsilon})\) time algorithm can exist that \((5/3 - \varepsilon)\)-approximates (for \(\varepsilon, \varepsilon' > 0\)) the eccentricities of all vertices in sparse graphs [8]. Better approximation algorithms are known for some special classes of graphs [27,34,35,42,43,50,51,94]. A number of heuristics for approximating diameters, radii and eccentricities in real-world graphs were proposed and investigated in [10,21,22,23,69,24,52].

Approximability of APSP is also extensively investigated. An additive 2-approximation for APSP in unweighted undirected graphs (the graphs we consider in this paper) was presented in [46]. It runs in \(O(\min\{n^{3/2}m^{1/2}, n^{7/3}\})\) time and hence improves the runtime of an earlier algorithm from [12]. In [19], an \(O(n^2)\) time algorithm was designed which computes an approximation of all distances with a multiplicative error of 2 and an additive error of 1. Furthermore, [19] gives an \(O(n^{2.24+o(1)}\varepsilon^{-3}\log(n/\varepsilon))\) time algorithm that computes an approximation of all distances with a multiplicative error of \((1 + \varepsilon)\) and an additive error of 2. The latter improves an earlier algorithm from [58]. Better algorithms are known for some special classes of graphs (see [25,35,49,89] and papers cited therein).

The need for fast approximation algorithms for estimating diameters, radii, centrality indices, or all pairs shortest paths in large-scale complex networks dictates to look for geometric and topological properties of those networks and utilize them algorithmically. The classical relationships between the diameter, radius, and center of trees and folklife linear time algorithms for their computation is one of the departing points of this research. A result from 1869 by C. Jordan [66] asserts that the radius of a tree \(T\) is roughly equal to half of its diameter and the center is either the middle vertex or the middle edge of any diametral path. The diameter and a diametral pair of \(T\) can be computed (in linear time) by a simple but elegant procedure: pick any vertex \(x\), find any vertex \(y\) furthest from \(x\), and find once more a vertex \(z\) furthest from \(y\); then return \(\{y, z\}\) as a diametral pair. This computation of a furthest vertex is called an \(FP\) scan; hence the diameter of a tree can be computed via two \(FP\) scans. This \(two\ FP\ scans\) procedure can be extended to exact or approximate computation of the diameter and radius in many classes of tree-like graphs. For example, this approach was used to compute the radius and a central vertex of a chordal graph in linear time [33]. In this case, the center of \(G\) is still close to the middle of all \((y, z)\)-shortest paths and \(d_{\text{G}}(y, z)\) is not the diameter but is still its good approximation: \(d(y, z) \geq \text{diam}(G) - 2\). Even better, the diameter of any chordal graph can be approximated in linear time with an additive error 1 [53]. But it turns out that the exact computation of diameters of chordal graphs is as difficult as the general diameter problem: it is even difficult to decide if the diameter of a split graph is 2 or 3.

The experience with chordal graphs shows that one have to abandon the hope of having fast exact algorithms, even for very simple (from metric point of view) graph-classes, and to search for fast algorithms approximating \(\text{diam}(G), \text{rad}(G), C(G), \text{eccg}(v)\) with a small additive constant depending only of the coarse geometry of the graph. Gromov hyperbolicity or the negative curvature of a graph (and, more generally, of a metric space) is one such constant. A graph \(G = (V, E)\) is \(\delta\)-hyperbolic [14,19,28,60] if for any four vertices \(w, v, x, y\) of \(G\), the two largest of the three distance sums \(d(w, v) + d(x, y), d(w, x) + d(v, y), d(w, y) + d(v, x)\) differ by at most \(2\delta \geq 0\). The hyperbolicity \(\delta(G)\) of a graph \(G\) is the smallest number \(\delta\) such that \(G\) is \(\delta\)-hyperbolic. The hyperbolicity can be viewed as a local measure of how close a graph is metrically to a tree:
the smaller the hyperbolicity is, the closer its metric is to a tree-metric (trees are 0-hyperbolic and chordal graphs are 1-hyperbolic).

Recent empirical studies showed that many real-world graphs (including Internet application networks, web networks, collaboration networks, social networks, biological networks, and others) are tree-like from a multi-ple point of view [10,11,20] or have small hyperbolicity [6,7,75,85]. It has been suggested in [77], and recently formally proved in [39], that the property, observed in real-world networks, in which traffic between nodes tends to go through a relatively small core of the network, as if the shortest paths between them are curved inwards, is due to the hyperbolicity of the network. Bending property of the eccentricity function in hyperbolic graphs were used in [16,15] to identify core-periphery structures in biological networks. Small hyperbolicity in real-world graphs provides many algorithmic advantages. Efficient approximate solutions are attainable for a number of optimization problems [35,36,37,39,40,44,57,92].

In [35] we initiated the investigation of diameter, center, and radius problems for \( \delta \)-hyperbolic graphs and we showed that the existing approach for trees can be extended to this general framework. Namely, it is shown in [35] that if \( G \) is a \( \delta \)-hyperbolic graph and \( \{y, z\} \) is the pair returned after two FP scans, then
\[
d(y, z) \geq \text{diam}(G) - 2\delta, \quad \text{diam}(G) \geq 2\text{rad}(G) - 4\delta - 1, \quad \text{diam}(C(G)) \leq 4\delta + 1,
\]
and \( C(G) \) is contained in a small ball centered at a middle vertex of any shortest \((y, z)\)-path. Consequently, we obtained linear time algorithms for the diameter and radius problems with additive errors linearly depending on the input graph’s hyperbolicity. In this paper, we advance this line of research and provide a linear time algorithm for approximate computation of the eccentricities (and thus of centrality indices) of all vertices of a \( \delta \)-hyperbolic graph \( G \), i.e., we compute the approximate values of all eccentricities within the same time bounds as one computes the approximation of the largest or the smallest eccentricity \((\text{diam}(G) \text{ or } \text{rad}(G))\). Namely, the algorithm outputs for every vertex \( v \) of \( G \) an estimate \( \hat{e}(v) \) of \( \text{ecc}_G(v) \) such that \( \text{ecc}_G(v) \leq \hat{e}(v) \leq \text{ecc}_G(v) + c\delta \), where \( c > 0 \) is a small constant. In fact, we demonstrate that \( G \) has a shortest path tree, constructible in linear time, such that for every vertex \( v \) of \( G \), \( \text{ecc}_G(v) \leq \text{ecc}_T(v) \leq \text{ecc}_G(v) + c\delta \) (a so-called eccentricity \( c\delta \)-approximating spanning tree). This is our first main result of this paper and the main ingredient in proving it is the following interesting dependency between the eccentricities of vertices of \( G \) and their distances to the center \( C(G) \): up to an additive error linearly depending on \( \delta \), \( \text{ecc}_G(v) \) is equal to \( d(v, C(G)) + \text{rad}(G) \). To establish this new result, we have to revisit the results of [35] about diameters, radii, and centers, by simplifying their proofs and extending them to all eccentricities.

Eccentricity \( k \)-approximating spanning trees were introduced by Prisner in [81]. A spanning tree \( T \) of a graph \( G \) is called an eccentricity \( k \)-approximating spanning tree if for every vertex \( v \) of \( G \) \( \text{ecc}_T(v) \leq \text{ecc}_G(v) + k \) holds [81]. Prisner observed that any graph admitting an additive tree \( k \)-spanner (that is, a spanning tree \( T \) such that \( d_T(v, u) \leq d_G(v, u) + k \) for every pair \( u, v \)) admits also an eccentricity \( k \)-approximating spanning tree. Therefore, eccentricity \( k \)-approximating spanning trees exist in interval graphs for \( k = 2 \) [70,75,80], in asteroidal-triple-free graph [70], strongly chordal graphs [20] and dual chordal graphs [20] for \( k = 3 \). On the other hand, although for every \( k \) there is a chordal graph without an additive tree \( k \)-spanner [70,80], yet as Prisner demonstrated in [81], every chordal graph has an eccentricity 2-approximating spanning tree. Later this result was extended in [81] to a larger family of graphs which includes all chordal graphs and all plane triangulations with inner vertices of degree at least 7. Both those classes belong to the class of 1-hyperbolic graphs. Thus, our result extends the result of [81] to all \( \delta \)-hyperbolic graphs.

As our second main result, we show that in every \( \delta \)-hyperbolic graph \( G \) all distances with an additive one-sided error of at most \( c'\delta \) can be found in \( O(|V|^2 \log^2 |V|) \) time, where \( c' < c \) is a small constant. With a recent result in [23], this demonstrates an equivalence between approximating the hyperbolicity and approximating the distances in graphs. Note that every \( \delta \)-hyperbolic graph \( G \) admits a distance approximating tree \( T \) [35,36,37], that is, a tree \( T \) (which is not necessarily a spanning tree) such that
\[
d_T(v, u) \leq d_G(v, u) + O(\delta \log n)
\]
for every pair \( u, v \). Such a tree can be used to compute all distances in \( G \) with an additive one-sided error of at most \( O(\delta \log n) \) in \( O(|V|^2) \) time. Our new result removes the dependency of the additive error from \( \log n \) and has a much smaller constant in front of \( \delta \). Note also that the tree \( T \) may use edges not present in \( G \) (not a spanning tree of \( G \)) and thus cannot serve as an eccentricity \( O(\delta \log n) \)-approximating spanning tree. Furthermore, as chordal graphs are 1-hyperbolic, for every \( k \) there is a 1-hyperbolic graph without an additive tree \( k \)-spanner [70,80].
At the conclusion of this paper, we analyze the performance of our algorithms for approximating eccentricities and distances on a number of real-world networks. Our experimental results show that the estimates on eccentricities and distances obtained are even better than the theoretical bounds proved.

2 Preliminaries

2.1 Center, diameter, centrality

All graphs \( G = (V, E) \) occurring in this paper are finite, undirected, connected, without loops or multiple edges. We use \( n \) and \( |V| \) interchangeably to denote the number of vertices and \( m \) and \( |E| \) to denote the number of edges in \( G \). The length of a path from a vertex \( v \) to a vertex \( u \) is the number of edges in the path. The distance \( d_G(u, v) \) between vertices \( u \) and \( v \) is the length of a shortest path connecting \( u \) and \( v \) in \( G \). The eccentricity of a vertex \( v \), denoted by \( ecc_G(v) \), is the largest distance from \( v \) to any other vertex, i.e., \( ecc_G(v) = \max_{w \in V} d_G(v, w) \). The centrality index of a vertex \( v \) is \( \frac{1}{|V|} \sum_{u \in V} d_G(u, v) \). The radius \( rad(G) \) of a graph \( G \) is the minimum eccentricity of a vertex in \( G \), i.e., \( rad(G) = \min_{v \in V} ecc_G(v) \). The diameter \( diam(G) \) of a graph \( G \) is the maximum eccentricity of a vertex in \( G \), i.e., \( diam(G) = \max_{v \in V} ecc_G(v) \). The center \( C(G) = \{ v \in V : ecc_G(v) = rad(G) \} \) of a graph \( G \) is the set of vertices with minimum eccentricity.

2.2 Gromov hyperbolicity and thin geodesic triangles

Let \( (X, d) \) be a metric space. The Gromov product of \( y, z \in X \) with respect to \( w \) is defined to be

\[
(y|z)_w = \frac{1}{2}(d(y, w) + d(z, w) - d(y, z)).
\]

A metric space \((X, d)\) is said to be \( \delta \)-hyperbolic \([60]\) for \( \delta \geq 0 \) if

\[
(x|y)_w \geq \min\{(x|z)_w, (y|z)_w\} - \delta
\]

for all \( w, x, y, z \in X \). Equivalently, \((X, d)\) is \( \delta \)-hyperbolic if for any four points \( u, v, x, y \) of \( X \), the two largest of the three distance sums \( d(u, v) + d(x, y), d(u, x) + d(v, y), d(u, y) + d(v, x) \) differ by at most \( 2\delta \geq 0 \). A connected graph \( G = (V, E) \) is \( \delta \)-hyperbolic (or of hyperbolicity \( \delta \)) if the metric space \((V, d_G)\) is \( \delta \)-hyperbolic, where \( d_G \) is the standard shortest path metric defined on \( G \).

\( \delta \)-Hyperbolic graphs generalize \( k \)-chordal graphs and graphs of bounded tree-length: each \( k \)-chordal graph has the tree-length at most \( \frac{k}{2} \) \([17]\) and each tree-length \( \lambda \) graph has hyperbolicity at most \( \lambda \) \([35,36]\). Recall that a graph is \( k \)-chordal if its induced cycles are of length at most \( k \), and it is of tree-length \( \lambda \) if it has a Robertson-Seymour tree-decomposition into bags of diameter at most \( \lambda \) \([37]\).

For geodesic metric spaces and graphs there exist several equivalent definitions of \( \delta \)-hyperbolicity involving different but comparable values of \( \delta \) \([12,28,59,160]\). In this paper, we will use the definition via thin geodesic triangles. Let \((X, d)\) be a metric space. A geodesic joining two points \( x \) and \( y \) from \( X \) is a (continuous) map \( f \) from the segment \([a, b]\) of \( \mathbb{R}^1 \) of length \( |a - b| = d(x, y) \) to \( X \) such that \( f(a) = x, f(b) = y \), and \( d(f(s), f(t)) = |s - t| \) for all \( s, t \in [a, b] \). A metric space \((X, d)\) is geodesic if every pair of points in \( X \) can be joined by a geodesic. Every unweighted graph \( G = (V, E) \) equipped with its standard distance \( d_G \) can be transformed into a geodesic (network-like) space \((X, d)\) by replacing every edge \( e = uv \) by a segment \([u, v]\) of length \( 1 \); the segments may intersect only at common ends. Then \((V, d_G)\) is isometrically embedded in a natural way in \((X, d)\). The restrictions of geodesics of \( X \) to the vertices \( V \) of \( G \) are the shortest paths of \( G \).

Let \((X, d)\) be a geodesic metric space. A geodesic triangle \( \Delta(x, y, z) \) with \( x, y, z \in X \) is the union \([x, y] \cup [x, z] \cup [y, z]\) of three geodesic segments connecting these vertices. Let \( m_x \) be the point of the geodesic segment \([y, z]\) located at distance \( \alpha_x := (y|z)_x = (d(y, x) + d(y, z) - d(x, z))/2 \) from \( y \). Then \( m_x \) is located at distance \( \alpha_z := (y|x)_z = (d(z, y) + d(z, x) - d(y, x))/2 \) from \( z \) because \( \alpha_y + \alpha_z = d(y, z) \). Analogously, define the points \( m_y \in [x, z] \) and \( m_z \in [x, y] \) both located at distance \( \alpha_x := (y|z)_x = (d(x, y) + d(x, z) - d(y, z))/2 \) from \( x \); see Fig.\( 1 \) for an illustration. There exists a unique isometry \( \varphi \) which maps \( \Delta(x, y, z) \) to a tripod \( T(x, y, z) \) consisting of three solid segments \([x, m], [y, m], \text{ and } [z, m]\) of lengths \( \alpha_x, \alpha_y, \) and \( \alpha_z \), respectively. This isometry maps the vertices \( x, y, z \) of \( \Delta(x, y, z) \) to the respective leaves of \( T(x, y, z) \) and the points
$m_x, m_y,$ and $m_z$ to the center $m$ of this tripod. Any other point of $T(x, y, z)$ is the image of exactly two points of $\Delta(x, y, z)$. A geodesic triangle $\Delta(x, y, z)$ is called $\delta$-thin if for all points $u, v \in \Delta(x, y, z)$, $\varphi(u) = \varphi(v)$ implies $d(u, v) \leq \delta$. A graph $G = (V, E)$ whose all geodesic triangles $\Delta(u, v, w)$, $u, v, w \in V$, are $\delta$-thin is called a graph with $\delta$-thin triangles, and $\delta$ is called the thinness parameter of $G$.

The following result shows that hyperbolicity of a geodesic space or a graph is equivalent to having thin geodesic triangles.

**Proposition 1** ([14, 28, 59, 60]). Geodesic triangles of geodesic $\delta$-hyperbolic spaces or graphs are $4\delta$-thin. Conversely, geodesic spaces or graphs with $\delta$-thin triangles are $\delta$-hyperbolic.

In what follows, we will need few more notions and notations. Let $G = (V, E)$ be a graph. By $[x, y]$ we denote a shortest path connecting vertices $x$ and $y$ in $G$; we call $[x, y]$ a geodesic between $x$ and $y$. A ball $B(s, r)$ of $G$ centered at vertex $s \in V$ and with radius $r$ is the set of all vertices with distance no more than $r$ from $s$ (i.e., $B(s, r) := \{v \in V : d_G(v, s) \leq r\}$). The $k$th-power of a graph $G = (V, E)$ is the graph $G^k = (V, E')$ such that $xy \in E'$ if and only if $0 < d_G(x, y) \leq k$. Denote by $F(x) := \{y \in V : d_G(x, y) = ecc_G(x)\}$ the set of all vertices of $G$ that are most distant from $x$. Vertices $x$ and $y$ of $G$ are called mutually distant if $x \in F(y)$ and $y \in F(x)$, i.e., $ecc_G(x) = ecc_G(y) = d_G(x, y)$.

### 3 Fast approximation of eccentricities

In this section, we give linear and almost linear time algorithms for sharp estimation of the diameters, the radii, the centers and the eccentricities of all vertices in graphs with $\delta$-thin triangles. Before presenting those algorithms, we establish some conditional lower bounds on complexities of computing the diameters and the radii in those graphs.

#### 3.1 Conditional lower bounds on complexities

Recent work has revealed convincing evidence that solving the diameter problem in subquadratic time might not be possible, even in very special classes of graphs. Roditty and Vassilevska W. [83] showed that an algorithm that can distinguish between diameter 2 and 3 in a sparse graph in subquadratic time refutes the following widely believed conjecture.

**The Orthogonal Vectors Conjecture:** There is no $\epsilon > 0$ such that for all $c \geq 1$, there is an algorithm that given two lists of $n$ binary vectors $A, B \subseteq \{0, 1\}^d$ where $d = c \log n$ can determine if there is an orthogonal pair $a \in A, b \in B$, in $O(n^{2-\epsilon})$ time.

Williams [95] showed that the Orthogonal Vectors (OV) Conjecture is implied by the well-known Strong Exponential Time Hypothesis (SETH) of Impagliazzo, Paturi, and Zane [64, 63]. Nowadays many papers base the hardness of problems on SETH and the OV conjecture (see, e.g., [82, 91] and papers cited therein).

Since all geodesic triangles of a graph constructed in the reduction in [83] are 2-thin, we can rephrase the result from [83] as follows.

![Fig. 1. A geodesic triangle $\Delta(x, y, z)$, the points $m_x, m_y, m_z$, and the tripod $T(x, y, z)$](image)
Statement 1 If for some $\epsilon > 0$, there is an algorithm that can determine if a given graph with $2$-thin triangles, $n$ vertices and $m = O(n)$ edges has diameter $2$ or $3$ in $O(n^{2-\epsilon})$ time, then the Orthogonal Vector Conjecture is false.

To prove a similar lower bound result for the radius problem, recently Abboud et al. [S] suggested to use the following natural and plausible variant of the OV conjecture.

The Hitting Set Conjecture: There is no $\epsilon > 0$ such that for all $c \geq 1$, there is an algorithm that given two lists $A, B$ of $n$ subsets of a universe $U$ of size $c \log n$, can decide in $O(n^{2-\epsilon})$ time if there is a set in the first list that intersects every set in the second list, i.e. a hitting set.

Abboud et al. [S] showed that an algorithm that can distinguish between radius $2$ and $3$ in a sparse graph in subquadratic time refutes the Hitting Set Conjecture. Since all geodesic triangles of a graph constructed in the reduction in [S] are $2$-thin, rephrasing that result from [S], we have,

Statement 2 If for some $\epsilon > 0$, there is an algorithm that can determine if a given graph with $2$-thin triangles, $n$ vertices, and $m = O(n)$ edges has radius $2$ or $3$ in $O(n^{2-\epsilon})$ time, then the Hitting Set Conjecture is false.

3.2 Fast additive approximations

In this subsection, we show that in a graph $G$ with $\delta$-thin triangles the eccentricities of all vertices can be computed in total linear time with an additive error depending on $\delta$. We establish that the eccentricity of a vertex is determined (up-to a small error) by how far the vertex is from the center $C(G)$ of $G$. Finally, we show how to construct a spanning tree $T$ of $G$ in which the eccentricity of any vertex is its eccentricity in $G$ up to an additive error depending only on $\delta$. For these purposes, we revisit and extend several results from our previous paper [55] concerning the linear time approximation of diameter, radius, and centers of $\delta$-hyperbolic graphs. For these particular cases, we provide simplified proofs, leading to better additive errors due to the use of thinness of triangles instead of the four point condition and to the computation in $O(\delta|E|)$ time of a pair of mutually distant vertices.

Define the eccentricity layers of a graph $G$ as follows: for $k = 0, \ldots, \text{diam}(G) - \text{rad}(G)$ set

$$C^k(G) := \{v \in V : \text{ecc}_G(v) = \text{rad}(G) + k\}.$$ 

With this notation, the center of a graph is $C(G) = C^0(G)$. In what follows, it will be convenient to define also the eccentricity of the middle point $m$ of any edge $xy$ of $G$; set $\text{ecc}_G(m) = \min\{\text{ecc}_G(x), \text{ecc}_G(y)\} + 1/2$.

We start with a proposition showing that, in a graph $G$ with $\delta$-thin triangles, a middle vertex of any geodesic between two mutually distant vertices has the eccentricity close to $\text{rad}(G)$ and is not too far from the center $C(G)$ of $G$.

Proposition 2. Let $G$ be a graph with $\delta$-thin triangles, $u, v$ be a pair of mutually distant vertices of $G$.

(a) If $c^*$ is the middle point of any $(u, v)$-geodesic, then $\text{ecc}_G(c^*) \leq \frac{d_G(u,v)}{2} + \delta \leq \text{rad}(G) + \delta$.
(b) If $c$ is a middle vertex of any $(u, v)$-geodesic, then $\text{ecc}_G(c) \leq [\frac{d_G(u,v)}{2}] + \delta \leq \text{rad}(G) + \delta$.
(c) $d_G(u,v) \geq 2\text{rad}(G) - 2\delta - 1$. In particular, $\text{diam}(G) \geq 2\text{rad}(G) - 2\delta - 1$.
(d) If $c$ is a middle vertex of any $(u, v)$-geodesic and $x \in C^k(G)$, then $k - \delta \leq d_G(x, c) \leq k + 2\delta + 1$. In particular $C(G) \subseteq B(c, 2\delta + 1)$.

Proof. Let $x$ be an arbitrary vertex of $G$ and $\Delta(u, v, x) := [u, v] \cup [v, x] \cup [x, u]$ be a geodesic triangle, where $[v, x], [x, u]$ are arbitrary geodesics connecting $x$ with $v$ and $u$. Let $m_x$ be a point on $[u, v]$ which is at distance $(x|v)_u = \frac{1}{2}(d(x, v) + d(v, u) - d(x, u))$ from $v$ and hence at distance $(x|u)_v = \frac{1}{2}(d(x, u) + d(v, u) - d(x, v))$ from $u$. Since $u$ and $v$ are mutually distant, we can assume, without loss of generality, that $c^*$ is located on $[u, v]$ between $v$ and $m_x$, i.e., $d(v, c^*) \leq d(v, m_x) = (x|u)_v$, and hence $(x|v)_u \leq (x|v)_u$. Since $d_G(v, x) \leq d_G(v, u)$, we also get $(u|v)_x \leq (x|v)_u$. 

6
(a) By the triangle inequality and since $d_G(u, v) \leq \text{diam}(G) \leq 2\text{rad}(G)$, we get
\[
d_G(x, c^*) \leq e(u|v)_x + \delta + d_G(u, c^*) - (x|v)_u \\
\leq d_G(u, c^*) + \delta = \frac{d_G(u, v)}{2} + \delta \leq \text{rad}(G) + \delta.
\]

(b) Since $c^* = c$ when $d_G(u, v)$ is even and $d_G(c^*, c) = \frac{1}{2}$ when $d_G(u, v)$ is odd, we have $\text{ecc}(c) \leq \text{ecc}(c^*) + \frac{1}{2}$. Additionally to the proof of (a), one needs only to consider the case when $d_G(u, v)$ is odd. We know that the middle point $c^*$ sees all vertices of $G$ within distance at most $\frac{d_G(u, v)}{2} + \delta$. Hence, both ends of the edge of $(u, v)$-geodesic, containing the point $c^*$ in the middle, have eccentricities at most
\[
\frac{d_G(u, v)}{2} + 1 + \delta = \left(\frac{d_G(u, v)}{2}\right) + \delta \leq \frac{2\text{rad}(G) - 1}{2} + \delta = \text{rad}(G) + \delta.
\]

(c) Since a middle vertex $c$ of any $(u, v)$-geodesic sees all vertices of $G$ within distance at most $\left(\frac{d_G(u, v)}{2}\right) + \delta$, if $d_G(u, v) \leq 2\text{rad}(G) - 2\delta - 2$, then
\[
\text{ecc}_G(c) \leq \left(\frac{d_G(u, v)}{2}\right) + \delta \leq \frac{2\text{rad}(G) - 2\delta - 2}{2} + \delta < \text{rad}(G),
\]
which is impossible.

(d) In the proof of (a), instead of an arbitrary vertex $x$, consider any vertex $x$ from $C^k(G)$. By the triangle inequality and since $d_G(u, v) \geq 2\text{rad}(G) - 2\delta - 1$ and both $d_G(u, x)$, $d_G(x, v)$ are at most $\text{rad}(G) + k$, we get
\[
d_G(x, c^*) \leq (u|v)_x + \delta + (x|u)_v - d_G(v, c^*) = d_G(v, x) - d_G(v, c^*) + \delta \\
\leq \text{rad}(G) + k - \frac{d_G(u, v)}{2} + \delta \leq k + 2\delta + 1.
\]
Consequently, $d_G(x, c) \leq d_G(x, c^*) + \frac{1}{2} \leq k + 2\delta + 1$. On the other hand, since $\text{ecc}_G(x) \leq \text{ecc}_G(c) + d_G(x, c)$ and $\text{ecc}_G(c) \leq \text{rad}(G) + \delta$, by statement (a), we get
\[
d_G(x, c) \geq \text{ecc}_G(x) - \text{ecc}_G(c) = k + \text{rad}(G) - \text{ecc}_G(c) \\
\geq (k + \text{rad}(G)) - (\text{rad}(G) + \delta) = k - \delta.
\]

\[\square\]

As an easy consequence of Proposition 2(d), we get that the eccentricity $\text{ecc}_G(x)$ of any vertex $x$ is equal, up to an additive one-sided error of at most $4\delta + 2$, to $d_G(x, C(G)) + \text{rad}(G)$.

**Corollary 1.** For every vertex $x$ of a graph $G$ with $\delta$-thin triangles,
\[
d_G(x, C(G)) + \text{rad}(G) - 4\delta - 2 \leq \text{ecc}_G(x) \leq d_G(x, C(G)) + \text{rad}(G).
\]

**Proof.** Consider an arbitrary vertex $x$ in $G$ and assume that $\text{ecc}_G(x) = \text{rad}(G) + k$. Let $c_x$ be a vertex from $C(G)$ closest to $x$. By Proposition 2(d), $d_G(c_x, c_x) \leq 2\delta + 1$ and $d_G(x, c) \leq k + 2\delta + 1 = \text{ecc}_G(x) - \text{rad}(G) + 2\delta + 1$. Hence,
\[
d_G(x, C(G)) \geq d_G(x, c_x) \leq d_G(x, c) + d_G(c, c_x) \leq d_G(x, c) + 2\delta + 1
\]
and
\[
\text{ecc}_G(x) \geq d_G(x, c) + \text{rad}(G) - 2\delta - 1.
\]
Combining both inequalities, we get
\[
\text{ecc}_G(x) \geq d_G(x, C(G)) + \text{rad}(G) - 4\delta - 2.
\]
Note also that, by the triangle inequality, $\text{ecc}_G(x) \leq d_G(x, c_x) + \text{ecc}_G(c_x) = d_G(x, C(G)) + \text{rad}(G)$ (that is, the right-hand inequality holds for all graphs). \[\square\]
It is interesting to note that the equality $ecc_G(x) = d_G(x, C(G)) + rad(G)$ holds for every vertex of a graph $G$ if and only if the eccentricity function $ecc_G(\cdot)$ on $G$ is unimodal (that is, every local minimum is a global minimum)\cite{18}. A slightly weaker condition holds for all chordal graphs \cite{51}: for every vertex $x$ of a chordal graph $G$, $ecc_G(x) \geq d_G(x, C(G)) + rad(G) - 1$.

**Proposition 3.** Let $G$ be a graph with $\delta$-thin triangles and $u, v$ be a pair of vertices of $G$ such that $v \in F(u)$.

(a) If $w$ is a vertex of a $(u, v)$-geodesic at distance $rad(G)$ from $v$, then $ecc_G(w) \leq rad(G) + \delta$.

(b) For every pair of vertices $x, y \in V$, $\max\{d_G(x, v), d_G(y, v)\} \geq d_G(x, y) - 2\delta$.

(c) $ecc_G(v) \geq diam(G) - 2\delta \geq 2rad(G) - 4\delta - 1$.

(d) Consider any $(v, t)$-geodesic at distance $\lfloor \frac{d_G(v, t)}{2} \rfloor$ from $t$ and $x \in C^k(G)$, then $ecc_G(c) \leq rad(G) + 3\delta$ and $k - 3\delta \leq d_G(x, c) \leq k + 3\delta + 1$. In particular, $C(G) \subseteq B(c, 3\delta + 1)$.

**Proof.** (a) Let $x$ be a vertex of $G$ with $d_G(x, w) = ecc_G(w)$. Let $\Delta(u, v, x) := [u, v] \cup [v, x] \cup [x, u]$ be a geodesic triangle, where $[v, x], [x, u]$ are arbitrary geodesics connecting $x$ with $v$ and $u$. Let $m_x$ be a point on $[u, v]$ which is at distance $(x|u)_v = \frac{1}{2}(d(x, v) + d(v, u) - d(x, u))$ from $v$ and hence at distance $(x|v)_u = \frac{1}{2}(d(x, u) + d(v, u) - d(x, v))$ from $u$. We distinguish between two cases: $w$ is between $u$ and $m_x$ or $w$ is between $v$ and $m_x$ in $[u, v]$.

In the first case, by the triangle inequality and $d_G(u, v) \leq d_G(u, v)$ (and hence, $(u|x)_v \geq (u|v)_x$), we get

$$d_G(w, x) \leq rad(G) - (u|x)_v + \delta + (u|v)_x \leq rad(G) + \delta.$$

In the second case, by the triangle inequality and since $d_G(v, x) \leq diam(G) \leq 2rad(G)$, we get

$$d_G(w, x) \leq (u|x)_v - rad(G) + \delta + (u|v)_x$$

$$\leq d_G(x, v) - rad(G) + \delta$$

$$\leq 2rad(G) - rad(G) + \delta = rad(G) + \delta.$$

(b) Consider an arbitrary $(u, v)$-geodesic $[u, v]$. Let $\Delta(u, x, v) := [u, v] \cup [v, x] \cup [x, u]$ be a geodesic triangle, where $[v, x], [x, u]$ are arbitrary geodesics connecting $x$ with $v$ and $u$. Let $\Delta(u, x, v) := [u, v] \cup [v, y] \cup [y, u]$ be a geodesic triangle, where $[v, y], [y, u]$ are arbitrary geodesics connecting $y$ with $v$ and $u$.

Let $m_x$ be a point on $[u, v]$ which is at distance $(x|u)_v = \frac{1}{2}(d(x, v) + d(v, u) - d(x, u))$ from $v$ and hence at distance $(x|v)_u = \frac{1}{2}(d(x, u) + d(v, u) - d(x, v))$ from $u$. We distinguish between two cases: $w$ is between $v$ and $m_x$ or $w$ is between $u$ and $m_x$ in $[u, v]$.

Since $d_G(u, v) \geq d_G(u, x)$ (as $v \in F(u)$), we have $(u|x)_v \leq (u|x)_v$. By the triangle inequality, we get

$$d_G(x, y) \leq (u|v)_x + \delta + ((y|u)_v - (u|x)_v) + \delta + (u|v)_y$$

$$\leq (u|x)_v - (u|x)_v + 2\delta + (y|u)_v + (u|v)_y$$

$$= d_G(v, y) + 2\delta.$$

Consequently, $\max\{d_G(v, x), d_G(v, y)\} \geq d_G(v, y) \geq d_G(x, y) - 2\delta$.

(c) Now, if $x, y$ is a diametral pair, i.e., $d_G(x, y) = diam(G)$, then, by (b) and Proposition 2(c),

$$ecc_G(v) \geq \max\{d_G(v, x), d_G(v, y)\}$$

$$\geq d_G(x, y) - 2\delta = diam(G) - 2\delta$$

$$\geq 2rad(G) - 4\delta - 1.$$

(d) Consider any $(v, t)$-geodesic $[v, t]$ and let $c^*$ be the middle point of it, $w$ be a vertex of $[v, t]$ at distance $rad(G)$ from $t$, and $c$ be a vertex of $[v, t]$ at distance $\lfloor \frac{d_G(v, t)}{2} \rfloor$ from $t$. We know by (a) that $ecc_G(w) \leq rad(G) + \delta$. Furthermore, since $2rad(G) \geq d_G(v, t) \geq 2rad(G) - 4\delta - 1$ (by (c)), $rad(G) \geq d_G(t, c) = \lfloor \frac{d_G(v, t)}{2} \rfloor \geq rad(G) - 2\delta$. Hence,

$$d_G(w, c) = d_G(w, t) - d_G(c, t) \leq rad(G) - rad(G) + 2\delta = 2\delta,$$

8
implying
\[ ecc_G(c) \leq d_G(w, c) + ecc_G(w) \leq \text{rad}(G) + 3\delta. \]

Let now \( x \) be an arbitrary vertex from \( C^k(G) \), i.e., \( ecc_G(x) \leq \text{rad}(G) + k \), for some integer \( k \geq 0 \). Consider a geodesic triangle \( \Delta(t, v, x) := [t, v] \cup [v, x] \cup [x, t] \), where \( [v, x], [x, t] \) are arbitrary geodesics connecting \( x \) with \( t \) and \( v \). Let \( m_x \) be a point on \([t, v]\) which is at distance \((x|t) = \frac{1}{2}(d(x, t) + d(v, t) - d(x, v))\) from \( v \) and hence at distance \((x|v) = \frac{1}{2}(d(x, t) + d(v, t) - d(x, t))\) from \( t \). Since, in what follows, we will use only the fact that \( d_G(v, t) \geq 2\text{rad}(G) - 4\delta - 1 \), we can assume, without loss of generality, that \( c^* \) is located on \([t, v]\) between \( v \) and \( m_x \), i.e., \( d(v, c^*) \leq d(v, m_x) = (x|t)_v \).

By the triangle inequality and since \( d_G(v, t) \geq 2\text{rad}(G) - 4\delta - 1 \) and both \( d_G(t, x) \) and \( d_G(x, v) \) are at most \( \text{rad}(G) + k \), we get
\[
d_G(x, c^*) \leq ((t|v)_x + \delta + (x|t)_v - d_G(v, c^*) = d_G(v, x) - d_G(v, c^*) + \delta
\leq \text{rad}(G) + k - \frac{d_G(v, t)}{2} + \delta \leq k + 3\delta + \frac{1}{2}.
\]
Hence, \( d_G(x, c) \leq d_G(x, c^*) + \frac{1}{2} \leq k + 3\delta + 1 \). On the other hand, since \( ecc_G(x) \leq ecc_G(c) + d_G(x, c) \) and \( ecc_G(c) \leq \text{rad}(G) + 3\delta \), we get
\[
d_G(x, c) \geq ecc_G(x) - ecc_G(c) = k + \text{rad}(G) - ecc_G(c)
\geq (k + \text{rad}(G)) - (\text{rad}(G) + 3\delta) = k - 3\delta.
\]

\[\square\]

**Proposition 4.** For every graph \( G \) with \( \delta \)-thin triangles, \( \text{diam}(C^k(G)) \leq 2k + 2\delta + 1 \). In particular, \( \text{diam}(C(G)) \leq 2\delta + 1 \).

**Proof.** Let \( x, y \) be two vertices of \( C^k(G) \) such that \( d_G(x, y) = \text{diam}(C^k(G)) \). Pick any \((x, y)\)-geodesic and consider the middle point \( m \) of it. Let \( z \) be a vertex of \( G \) such that \( d_G(m, z) = ecc_G(m) \). Consider a geodesic triangle \( \Delta(x, y, z) := [x, y] \cup [y, z] \cup [z, x] \), where \([x, z], [y, z] \) are arbitrary geodesics connecting \( z \) with \( x \) and \( y \). Let \( m_z \) be a point on \([x, y]\) which is at distance \((x|z)_y = \frac{1}{2}(d(x, y) + d(z, y) - d(x, z))\) from \( y \) and hence at distance \((y|z)_x = \frac{1}{2}(d(x, y) + d(z, x) - d(y, z))\) from \( x \). Without loss of generality, we can assume that \( m \) is located on \([x, y]\) between \( y \) and \( m_z \).

Since \( ecc_G(y) \leq \text{rad}(G) + k \), we have
\[
d_G(m, z) = ecc_G(m) \geq \text{rad}(G) - \frac{1}{2} \geq ecc_G(y) - k - \frac{1}{2} = d_G(y, z) - k - \frac{1}{2}.
\]

On the other hand, by the triangle inequality, we get
\[
d_G(m, z) \leq (x|z)_y - d_G(y, m) + \delta + (x|y)_z = d_G(y, z) - d_G(y, m) + \delta
\leq d_G(y, z) - \frac{d_G(x, y)}{2} + \delta.
\]
Hence, \( d_G(x, y) \leq 2k + 2\delta + 1 \).

\[\square\]

**Diameter and radius.** For an arbitrary connected graph \( G = (V, E) \) and a given vertex \( u \in V \), a most distant from \( u \) vertex \( v \in F(u) \) can be found in linear \( (O(|E|)) \) time by a breadth-first-search \( BFS(u) \) started at \( u \). A pair of mutually distant vertices of a connected graph \( G = (V, E) \) with \( \delta \)-thin triangles can be computed in \( O(\delta|E|) \) total time as follows. By Proposition 3(c), if \( v \) is a most distant from an arbitrary vertex \( u \) and \( t \) is a most distant from \( v \), then \( d(v, t) \geq \text{diam}(G) - 2\delta \). Hence, using at most \( O(\delta) \) breadth-first-searches, one can generate a sequence of vertices \( v := v_1, t := v_2, v_3, \ldots v_k \) with \( k \leq 2\delta + 2 \) such that each \( v_i \) is most distant from \( v_{i-1} \) (with, \( v_0 = u \)) and \( v_k, v_{k-1} \) are mutually distant vertices (the initial value \( d(v, t) \geq \text{diam}(G) - 2\delta \) can be improved at most \( 2\delta \) times).

Thus, by Proposition 3 and Proposition 8 we get the following additive approximations for the radius and the diameter of a graph with \( \delta \)-thin triangles.
Corollary 2. Let \( G = (V, E) \) be a graph with \( \delta \)-thin triangles.

1. There is a linear \( (O(|E|)) \) time algorithm which finds in \( G \) a vertex \( v \) with eccentricity at most \( \text{rad}(G) + 3\delta \) and a vertex \( c \) with eccentricity at least \( \text{diam}(G) - 2\delta \). Furthermore, \( C(G) \subseteq B(c, 3\delta + 1) \) holds.

2. There is an algorithm which in total almost linear \( (O(\delta|E|)) \) time algorithm which finds in \( G \) a vertex \( c \) with eccentricity at most \( \text{rad}(G) + \delta \). Furthermore, \( C(G) \subseteq B(c, 2\delta + 1) \) holds.

All eccentricities. In what follows, we will show that all vertex eccentricities of a graph with \( \delta \)-thin triangles can be also additively approximated in (almost) linear time.

Proposition 5. Let \( G \) be a graph with \( \delta \)-thin triangles.

(a) Let \( c \) be a middle vertex of any \((u, v)\)-geodesic between a pair \( u, v \) of mutually distant vertices of \( G \) and \( T \) is a BFS\((c)\)-tree of \( G \), then, for every vertex \( x \) of \( G \), \( \text{ecc}_G(x) \leq \text{ecc}_T(x) \leq \text{ecc}_G(x) + 3\delta + 1 \).

(b) If \( v \) is a most distant vertex from an arbitrary vertex \( u \), \( t \) is a most distant vertex from \( v \), \( c \) is a vertex of a \((v, t)\)-geodesic at distance \( \left\lceil \frac{d_G(v, t)}{2} \right\rceil \) from \( t \) and \( T \) is a BFS\((c)\)-tree of \( G \), then \( \text{ecc}_G(x) \leq \text{ecc}_T(x) \leq \text{ecc}_G(x) + 6\delta + 1 \).

Proof. (a) Let \( x \) be an arbitrary vertex of \( G \) and assume that \( \text{ecc}_G(x) = \text{rad}(G) + k \) for some integer \( k \geq 0 \). We know from Proposition 2(b) that \( \text{ecc}_G(c) \leq \text{rad}(G) + \delta \). Consequently, by Proposition 2(d), \( \text{d}_G(c, x) \leq k + 2\delta + 1 \). Since \( T \) is a BFS\((c)\)-tree, \( \text{d}_G(x, c) = \text{d}_T(x, c) \) and \( \text{ecc}_G(c) = \text{ecc}_T(c) \). Consider a vertex \( y \) in \( G \) such that \( \text{d}_T(x, y) = \text{ecc}_T(x) \). We have

\[
\text{ecc}_T(x) = \text{d}_T(x, y) \leq \text{d}_T(x, c) + \text{d}_T(c, y) \leq \text{d}_G(x, c) + \text{ecc}_T(c) = \text{d}_G(x, c) + \text{ecc}_G(c) \leq k + 2\delta + 1 + \text{rad}(G) + \delta = \text{rad}(G) + k + 3\delta + 1
\]

As \( T \) is a spanning tree of \( G \), evidently, also \( \text{ecc}_G(x) \leq \text{ecc}_T(x) \) holds.

(b) The proof is similar to the proof of (a); only, in this case, \( \text{ecc}_G(c) \leq \text{rad}(G) + 3\delta \) and \( \text{d}_G(c, x) \leq k + 3\delta + 1 \) holds for every \( x \in C^k(G) \) (by Proposition 2(d)).

A spanning tree \( T \) of a graph \( G \) is called an eccentricity \( k \)-approximating spanning tree if for every vertex \( v \) of \( G \) \( \text{ecc}_T(v) \leq \text{ecc}_G(v) + k \) holds \([41,43]\). Thus, by Proposition 5, we get.

Theorem 1. Every graph \( G = (V, E) \) with \( \delta \)-thin triangles admits an eccentricity \((3\delta + 1)\)-approximating spanning tree constructible in \( O(|V|) \) time and an eccentricity \((6\delta + 1)\)-approximating spanning tree constructible in \( O(|E|) \) time.

Theorem 1 generalizes recent results from \([41,43]\) that chordal graphs and some of their generalizations admit eccentricity \( 2 \)-approximating spanning trees.

Note that the eccentricities of all vertices in any tree \( T = (V, U) \) can be computed in \( O(|V|) \) total time. As we noticed already, it is a folklore by now that for trees the following facts are true:

(1) The center \( C(T) \) of any tree \( T \) consists of one vertex or two adjacent vertices.

(2) The center \( C(T) \) and the radius \( \text{rad}(T) \) of any tree \( T \) can be found in linear time.

(3) For every vertex \( v \in V \), \( \text{ecc}_T(v) = \text{d}_T(v, C(T)) + \text{rad}(T) \).

Hence, using \( BFS(C(T)) \) on \( T \) one can compute \( \text{d}_T(v, C(T)) \) for all \( v \in V \) in total \( O(|V|) \) time. Adding now \( \text{rad}(T) \) to \( \text{d}_T(v, C(T)) \), one gets \( \text{ecc}_T(v) \) for all \( v \in V \). Consequently, by Theorem 1, we get the following additive approximations for the vertex eccentricities in graphs with \( \delta \)-thin triangles.

Theorem 2. Let \( G = (V, E) \) be a graph with \( \delta \)-thin triangles.

(1) There is an algorithm which in total linear \( O(|E|) \) time outputs for every vertex \( v \in V \) an estimate \( \hat{e}(v) \) of its eccentricity \( \text{ecc}_G(v) \) such that \( \text{ecc}_G(v) \leq \hat{e}(v) \leq \text{ecc}_G(v) + 6\delta + 1 \).

(2) There is an algorithm which in total almost linear \( O(\delta|E|) \) time outputs for every vertex \( v \in V \) an estimate \( \hat{e}(v) \) of its eccentricity \( \text{ecc}_G(v) \) such that \( \text{ecc}_G(v) \leq \hat{e}(v) \leq \text{ecc}_G(v) + 3\delta + 1 \).
4 Fast Additive Approximation of All Distances

Here, we will show that if the δ-th power \(G^\delta\) of a graph \(G\) with \(\delta\)-thin triangles is known in advance, then the distances in \(G\) can be additively approximated (with an additive one-sided error of at most \(\delta + 1\)) in \(O(|V|^2)\) time. If \(G^\delta\) is not known, then the distances can be additively approximated (with an additive one-sided error of at most \(2\delta + 2\)) in almost quadratic time.

Our method is a generalization of an unified approach used in [49] to estimate (or compute exactly) all pairs shortest paths in such special graph families as \(k\)-chordal graphs, chordal graphs, AT-free graphs and many others. For example: all distances in \(k\)-δ error of at most 2 \(x\) and distances in \(k\)-δ for all vertices \(x\) and \(y\) can be found in \(O(|V|^2)\) time; all distances in chordal graphs with an additive one-sided error of at most \(k-1\) can be found in \(O(|V|^2)\) time; all pairs shortest path problem on a chordal graph \(G\) can be solved in \(O(|V|^2)\) time if \(G^2\) is known. Note that in chordal graph all geodesic triangles are 2-thin.

Let \(G = (V, E)\) be a graph with \(\delta\)-thin triangles. Pick an arbitrary start vertex \(s \in V\) and construct a BFS(s)-tree \(T\) of \(G\) rooted at \(s\). Denote by \(p_T(x)\) the parent and by \(h_T(x) = d_T(x, s) = d_G(x, s)\) the height of a vertex \(x\) in \(T\). Since we will deal only with one tree \(T\), we will omit the subscript \(T\). Let \(P_T(x, s) := (x_q, x_{q-1}, \ldots, x_1)\) and \(P_T(y, s) := (y_p, y_{p-1}, \ldots, y_1, s)\) be the paths of \(T\) connecting vertices \(x\) and \(y\) with the root \(s\). By \(s\) \(T\)(\(x, y; \lambda\)) we denote the largest index \(k\) such that \(d_G(x_k, y_k) \leq \lambda\) (the \(\lambda\) separation level). Our method is based on the following simple fact.

**Proposition 6.** For every vertices \(x\) and \(y\) of a graph \(G\) with \(\delta\)-thin triangles and any BFS-tree \(T\) of \(G\),

\[h_T(x) + h_T(y) - 2k - 1 \leq d_G(x, y) \leq h_T(x) + h_T(y) - 2k + d_G(x_k, y_k),\]

where \(k = s\) \(T\)(\(x, y; \delta\)).

**Proof.** By the triangle inequality, \(d_G(x, y) \leq d_G(x, x_k) + d_G(x_k, y_k) + d_G(y_k, y) = h_T(x) + h_T(y) - 2k + d_G(x_k, y_k)\). Consider now an arbitrary \((x, y)\)-geodesic \([x, y]\) in \(G\). Let \(\Delta(x, y, s) := [x, y] \cup [x, s] \cup [y, s]\) be a geodesic triangle, where \([x, s] = P_T(x, s)\) and \([y, s] = P_T(y, s)\). Since \(\Delta(x, y, s)\) is \(\delta\)-thin, \(s\) \(T\)(\(x, y; \delta\)) \(\geq (|x|y) - \frac{1}{\delta}\). Hence, \(h_T(x) - s\) \(T\)(\(x, y; \delta\)) \(\leq (s|x|y + \frac{1}{\delta}\) and \(h_T(y) - s\) \(T\)(\(x, y; \delta\)) \(\leq (s|y|x + \frac{1}{\delta}\). As \(d_G(x, y) = (s|x|y + (s|y|x)\), we get \(d_G(x, y) \geq h_T(x) - s\) \(T\)(\(x, y; \delta\)) \(+ h_T(y) - s\) \(T\)(\(x, y; \delta\)) \(- 1\).

Note that we may regard BFS(s) as having produced a numbering from \(n\) to 1 in decreasing order of the vertices in \(V\) where vertex \(s\) is numbered \(n\). As a vertex is placed in the queue by BFS(s), it is given the next available number. The last vertex visited is given the number 1. Let \(s = [v_1, v_2, \ldots, v_n = s]\) be a BFS(s)-ordering of the vertices of \(G\) and \(T\) be a BFS(s)-tree of \(G\) produced by a BFS(s). Let \(\sigma(x)\) be the number assigned to a vertex \(x\) in this BFS(s)-ordering. For two vertices \(x\) and \(y\), we write \(x < y\) whenever \(\sigma(x) < \sigma(y)\).

First, we will show that if \(G^\delta\) is known in advance (i.e., its adjacency matrix is given) for a graph \(G\) with \(\delta\)-thin triangles, then the distances in \(G\) can be additively approximated (with an additive one-sided error of at most \(\delta + 1\)) in \(O(|V|^2)\) time. We consider the vertices of \(G\) in the order \(\sigma\) from 1 to \(n\). For each current vertex \(x\) we show that the values \(\tilde{d}(x, y) := h_T(x) + h_T(y) - 2s\) \(T\)(\(x, y; \delta\)) \(+ \delta\) for all vertices \(y\) with \(y > x\) can be computed in \(O(|V|)\) total time. By Proposition 6

\[d_G(x, y) \leq \tilde{d}(x, y) \leq d_G(x, y) + \delta + 1.\]

The values \(\tilde{d}(x, y)\) for all \(y\) with \(y > x\) can be computed using the following simple procedure. We will omit the subscripts \(G\) and \(T\) if no ambiguities arise. Let also \(L_i = \{v \in V : d(v, s) = i\}\). In the procedure, \(S_u\) represents vertices of a subtree of \(T\) rooted at \(u\).

(01) set \(q := h_T(x)\)
(02) define a set \(S_u := \{u\}\) for each vertex \(u \in L_q, u > x\), and denote this family of sets by \(F\)
(03) for \(k = q\) downto 0
(04) let \(x_k\) be the vertex from \(L_k \cap P_T(x, s)\)
(05) for each vertex \(u \in L_k\) with \(u > x\)
(06) if \(d_G(u, x_k) \leq \delta\) (i.e., \(u = x_k\) or \(u\) is adjacent to \(x_k\) in \(G^\delta\)) then
For every $v \in S_u$

$$d(x, v) := h(x) + h(v) - 2k + \delta$$

and remove $S_u$ from $\mathcal{F}$

endfor

/* update $\mathcal{F}$ for the next iteration */

if $k > 0$ then

for each vertex $u \in L_{k-1}$

combine all sets $S_{u_1}, \ldots, S_{u_\ell}$ from $\mathcal{F}$ ($\ell \geq 0$), such that $p_T(u_1) = \ldots = p_T(u_\ell) = u$

into one new set $S_u := \{u\} \cup S_{u_1} \cup \ldots \cup S_{u_\ell}$  /* when $\ell = 0$, $S_u := \{u\}$ */

endfor

endfor

set also $\hat{d}(x, s) := h(x)$.

Thus, we have the following result.

**Theorem 3.** Let $G = (V, E)$ be a graph with $\delta$-thin triangles. Given $G^d$, all distances in $G$ with an additive one-sided error of at most $\delta + 1$ can be found in $O(|V|^2)$ time.

To avoid the requirement that $G^d$ is given in advance, we can use any known fast constant-factor approximation algorithm in the preprocessing step, the distances in a graph $G$ with $\delta$-thin triangles can be additively approximated with an additive one-sided error of at most $\delta + 1$ in $O(T(|V|) + |V|^2)$ time.

Although one can use any known fast constant-factor approximation algorithm in the preprocessing step, in what follows, we will demonstrate our idea using a fast approximation algorithm from [19]. It computes in $O(|V|^2 \log^2 |V|)$ total time for every pair $x, y$ a value $\hat{d}(x, y)$ such that

$$d_G(x, y) \leq \hat{d}(x, y) \leq 2d_G(x, y) + 1.$$

Assume that the values $\hat{d}(x, y)$, $x, y \in V$, are precomputed. By $s_l(x, y; \lambda)$ we denote now the largest index $k$ such that $\hat{d}(x_k, y_k) \leq \lambda$. We have

**Proposition 7.** For every vertices $x$ and $y$ of a graph $G$ with $\delta$-thin triangles, any integer $\rho \geq \delta$, and any BFS-tree $T$ of $G$,

$$h_T(x) + h_T(y) - 2k - 1 \leq d_G(x, y) \leq h_T(x) + h_T(y) - 2k + d_G(x_k, y_k),$$

where $k = s_l(x, y; 2\rho + 1)$.

**Proof.** The proof is identical to the proof of Proposition 7. One needs only to notice the following. In a geodesic triangle $\Delta(x, y, s) := [x, y] \cup [x, s] \cup [y, s]$ with $[x, y] = P_T(x, s) = (x_q, x_{q-1}, \ldots, x_1, s)$ and $[y, s] = P_T(y, s) = (y_p, y_{p-1}, \ldots, y_1, s)$, for each $i \leq (x|y)_s$, $d_G(x, y_i) \leq \delta \leq \rho$ and, hence, $\hat{d}(x, y_i) \leq 2\rho + 1$ holds.

Therefore, $s_l(x, y; 2\rho + 1) \geq (x|y)_s - \frac{1}{2}$.

Let $\rho$ be any integer greater than or equal to $\delta$. By replacing in our earlier procedure lines (06) and (08) with

$$\text{if } \hat{d}(u, x_k) \leq 2\rho + 1 \text{ then}$$

$$\text{set } \hat{d}(x, v) := h(x) + h(v) - 2k + 2\rho + 1 \text{ and remove } S_u \text{ from } \mathcal{F}$$

we will compute for each current vertex $x$ all values $\hat{d}(x, y) := h_T(x) + h_T(y) - 2s_l(x, y; 2\rho + 1) + 2\rho + 1, y > x$, in $O(|V|)$ total time. By Proposition 7,

$$d_G(x, y) \leq h_T(x) + h_T(y) - 2s_l(x, y; 2\rho + 1) + d_G(x_k, y_k)$$

$$\leq h_T(x) + h_T(y) - 2s_l(x, y; 2\rho + 1) + \hat{d}(x, y_k)$$

$$\leq h_T(x) + h_T(y) - 2s_l(x, y; 2\rho + 1) + 2\rho + 1$$

$$= \hat{d}(x, y)$$
and
\[
\hat{d}(x, y) = h_T(x) + h_T(y) - 2\tilde{s}_T(x, y; 2\rho + 1) + 2\rho + 1 \\
\leq d_G(x, y) + 2\rho + 2.
\]

Thus, we have the following result:

**Theorem 4.** Let \( G = (V, E) \) be a graph with \( \delta \)-thin triangles.

(a) If the value of \( \delta \) is known, then all distances in \( G \) with an additive one-sided error of at most \( 2\delta + 2 \) can be found in \( O(|V|^2 \log^2 |V|) \) time.

(b) If an approximation \( \rho \) of \( \delta \) such that \( \delta \leq \rho \leq a\delta + b \) is known (where \( a \) and \( b \) are constants), then all distances in \( G \) with an additive one-sided error of at most \( 2(a\delta + b + 1) \) can be found in \( O(|V|^2 \log^2 |V|) \) time.

The second part of Theorem 4 says that if an approximation of the thinness parameter of a graph \( G \) is given then all distances in \( G \) can be additively approximated in \( O(|V|^2 \log^2 |V|) \) time. Recently, it was shown in [32] that the following converse is true. From an estimate of all distances in \( G \) with an additive one-sided error of at most \( k \), it is possible to compute in \( O(|V|^2) \) time an estimation \( \rho^* \) of the thinness of \( G \) such that \( \delta \leq \rho^* \leq 8\delta + 12k + 4 \), proving a \( O(|V|^2) \)-equivalence between approximating the thinness and approximating the distances in graphs.

5 **Experimentation on Some Real-World Networks**

In this section, we analyze the performance of our algorithms for approximating eccentricities and distances on a number of real-world networks. Our experimental results show that the estimates on eccentricities and distances obtained are even better than the theoretical bounds described in Corollary 2 and Theorems 3.4

We apply our algorithms to six social networks, four email communication networks, four biological networks, six internet graphs, four peer-to-peer networks, three web networks, two product-co-purchasing networks, and four infrastructure networks. Most of the networks listed are part of the Stanford Large Network Dataset Collection (snap) and the Koblenz Network Collection (konect), and are available at [1] and [2]. Characteristics of these networks, such as the number of vertices and edges, the average degree, the radius and the diameter, are given in Table 1. The numbers listed in Table 1 are based on the largest connected component of each network, when the entire network is disconnected. We ignore the directions of the edges and remove all self-loops from each network. Additionally, in Table 1 for each network we report the size (as the number of vertices) of its center \( C(G) \). We also analyze the diameter and the connectivity of the center of each network. The diameter of the center \( diam_G(C(G)) \) is defined as the maximum distance between any two central vertices in the graph. In the last column of Table 1 we report the Gromov hyperbolicity \( \delta \) of majority of network\(^3\). Computing the hyperbolicity of a graph is computationally expensive; therefore, we provide the exact \( \delta \) values for the smaller networks (those with \( |V| \leq 30K \)) in our dataset (in some cases, the algorithm proposed in [4] was used). For some larger networks, the approximated \( \delta \)-hyperbolicity values listed in Table 1 are as reported in [67]. Most networks that we included in our dataset are hyperbolic. However, for comparison reasons, we included also a few infrastructure networks that are known to lack the hyperbolicity property.

5.1 **Estimation of Eccentricities**

Following Proposition 2 for each graph in our dataset, we found a pair \( u, v \) of mutually distant vertices. In column two of Table 2 we report on how many \( BFS \) sweeps of a graph were needed to locate \( u \) and \( v \). Interestingly, for almost all graphs (28 out 33) only two sweeps were sufficient. For four other graphs (including ROAD-PA network whose hyperbolicity is large) three sweeps were needed, and only for one graph (POWER-GRID network) we needed four sweeps.

\(^2\) All \( \delta \)-hyperbolicity values listed in Table 1 were computed using Gromov’s four-point condition definition. As mentioned in [69](69), geodesic triangles of geodesic \( \delta \)-hyperbolic spaces are \( 4\delta \)-thin.

\(^3\) For WEB-STANFORD and WEB-BERKSTAN, [67] gives 1.5 and 2, respectively, as estimates on the hyperbolicities. However, the sampling method they used seems to be not very accurate. According to [75], the hyperbolicities are at least 7 for both graphs.
In Table 1, we report for each graph $G$ the difference between $2\text{rad}(G)$ and $d_G(u,v)$. Proposition 2(c) says that the difference must be at most $2\delta + 1$, where $\delta$ is the thickness of geodesic triangles in $G$. Actually, for large number (27 out of 33) of graphs in our dataset, the difference is at most two. For the remaining two graphs, the difference equals one. In all graphs, $d_G(u,v) = \text{diam}(G)$ for 27 graphs in our dataset, including ROAD-PA network whose geodesic thickness is at most 196. For remaining six graphs $d_G(u,v) = \text{diam}(G) - 1$ holds.

We also analyzed the quality of a middle vertex $c$ of a randomly picked shortest path between mutually distant vertices $u$ and $v$. Proposition 2 states that $\text{ecc}_G(c)$ is close to $\text{rad}(G)$ and $c$ is not too far from the graph’s center $C(G)$. Table 2 lists the properties of the selected middle vertex $c$. In almost all graphs, vertex $c$ belongs to the center $C(G)$ or is at distance one or two from $C(G)$. Even in graphs with $\text{ecc}_G(c) - \text{rad}(G) > 2$, the value $\text{ecc}_G(c) - \text{rad}(G)$ is smaller than what is suggested by Proposition 2(b). It is also clear from Table 2 that $c$ is not too far from any vertex in $C(G)$ (look at the radius $i$ of the ball $B(c,i)$ required to include $C(G)$). In all graphs, $i$ is much smaller than $2\delta + 1$ (indicated in Proposition 2(d)).

Following Theorem 4 for each graph $G = (V,E)$ in our dataset, we constructed an arbitrary BFS(c)-tree $T_1 = (V,E)$, rooted at vertex $c$, and analyzed how well $T_1$ preserves or approximates the eccentricities of vertices in $G$. By Theorem 1, $\text{ecc}_G(v) \leq \text{ecc}_{T_1}(v) \leq \text{ecc}_G(v) + 3\delta + 1$ holds for every $v \in V$. In our experiments, for each graph $G$ and the constructed BFS(c)-tree $T_1$, we computed $k_{\text{max}} := \max_{v \in V}(\text{ecc}_{T_1}(v) - \text{ecc}_G(v))$ (maximum distortion) and $k_{\text{avg}} := \frac{1}{|V|} \sum_{v \in V} \text{ecc}_{T_1}(v) - \text{ecc}_G(v)$ (average distortion). For most graphs (see Table 2), the value of $k_{\text{max}}$ is small: $k_{\text{max}} = 0$ for one graph, $k_{\text{max}} = 2$ for eight graphs, $k_{\text{max}} = 3$ for nine graphs, $k_{\text{max}} = 4$ for four graphs, $k_{\text{max}} = 5$ for two graphs, and $k_{\text{max}} > 5$ for nine graphs. Also, the average
Table 2. Qualities of a pair of mutually distant vertices $u$ and $v$, of a middle vertex $c$ of a $(u, v)$-geodesic, and of a BFS($c$)-tree $T_1$ rooted at vertex $c$. "No. of BFS iterations" indicates how many breadth-first-search iterations were needed to obtain a pair of mutually distant vertices $u$ and $v$. For each vertex $x \in V$, $k(x) := ecc_{T_1}(x) - ecc_G(x)$. Also, $k_{\text{max}} := \max_{x \in V} k(x)$ and $k_{\text{avg}} := \frac{1}{n} \sum_{x \in V} k(x)$.

distortion $k_{\text{avg}}$ is much smaller than $k_{\text{max}}$ for all graphs. In fact, $k_{\text{avg}} < 3$ in all but five graphs (GNUTELLA-30, GNUTELLA-31, AMAZON-2, POWER-GRID, and ROAD-PA). In graphs with high $k_{\text{max}}$, close inspection reveals that only small percent of vertices achieve this maximum. For example, in graph WEB-STANFORD, $k_{\text{max}} = 28$ was only achieved by 17 vertices. The distributions of the values of $k(v) := ecc_{T_1}(v) - ecc_G(v)$ of all graphs are listed in Table 4 (see Appendix).

Similar experiments were performed following Proposition 6. For each graph $G$ in our dataset, we picked a random vertex $v \in V$ and a random vertex $v \in F(u)$. Then, we identified in a randomly picked $(u, v)$-geodesic a vertex $w$ at distance $rad(G)$ from $v$. We did not consider a vertex $c$ defined in Proposition 3(d) since, for majority of graphs in our dataset, $c$ will be a middle vertex of a geodesic between two mutually distant vertices, and working with $c$ we will duplicate previous experiments. Recall that for majority of our graphs (as found in our experiments) two BFS sweeps already identify a pair of mutually distant vertices. We know from Proposition 6 that $ecc_G(v) \geq diam(G) - 2\delta \geq 2rad(G) - 4\delta - 1$ and $ecc_G(w) \leq rad(G) + \delta$. Our experimental results are better than these theoretical bounds. In Table 4 we list eccentricities of $v$ and $w$ for each graph. In almost all graphs, the eccentricity of $v$ is equal to the diameter $diam(G)$. Only four graphs have $ecc_G(v) = diam(G) - 1$ and one graph (ROAD-PA) has $ecc_G(v) > diam(G) - 1$. Vertex $w$ is central for 21 graphs, has eccentricity equal to $rad(G) + 1$ for 10 graphs, has eccentricity equal to $rad(G) + 2$ for one graph, and only for one remaining graph (ROAD-PA network, which has large hyperbolicity) its eccentricity is equal to $rad(G) + 15$. It turns out also (see columns five and six of Table 2) that vertex $w$ either belongs to the center $C(G)$ or is very close to the center. The only exception is again ROAD-PA network where $2rad(G) - ecc_G(w) = 32$ and $d(w, C(G)) = 21$.
and with a third BF S all graphs. The distributions of the values of $k$

For each graph in the dataset, three values of $k$ for $sc$-ppi $T$

<table>
<thead>
<tr>
<th>Network</th>
<th>$ecc_G(v)$</th>
<th>$2rad(G) - ecc_G(v)$</th>
<th>$ecc_G(w)$</th>
<th>$d_G(w, C(G))$</th>
<th>$\min_i : B(w, i) \ge C(G)$</th>
<th>$k_{max}$</th>
<th>$k_{avg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUTCH-ELITE</td>
<td>22</td>
<td>2</td>
<td>12</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>2.434</td>
</tr>
<tr>
<td>FACEBOOK</td>
<td>8</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0.704</td>
</tr>
<tr>
<td>EVA</td>
<td>18</td>
<td>2</td>
<td>11</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0.572</td>
</tr>
<tr>
<td>SLASHDOT</td>
<td>11</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.88</td>
</tr>
<tr>
<td>LOANS</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2.033</td>
</tr>
<tr>
<td>TWITTER</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>1.821</td>
</tr>
<tr>
<td>EMAIL-VIRGILI</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1.932</td>
</tr>
<tr>
<td>EMAIL-ENRON</td>
<td>13</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.903</td>
</tr>
<tr>
<td>EMAIL-EU</td>
<td>14</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0.002</td>
</tr>
<tr>
<td>WIKITALK-CN</td>
<td>8</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1.791</td>
</tr>
<tr>
<td>CE-METABOLIC</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.349</td>
</tr>
<tr>
<td>SC-PPI</td>
<td>19</td>
<td>3</td>
<td>12</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>4.196</td>
</tr>
<tr>
<td>YEAST-PI</td>
<td>11</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2.558</td>
</tr>
<tr>
<td>HOMO-PI</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.612</td>
</tr>
<tr>
<td>AS-GRAFH-1</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.887</td>
</tr>
<tr>
<td>AS-GRAFH-2</td>
<td>11</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0.833</td>
</tr>
<tr>
<td>AS-GRAFH-3</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.312</td>
</tr>
<tr>
<td>ROUTEVIEW</td>
<td>10</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.329</td>
</tr>
<tr>
<td>AS-CANDA</td>
<td>17</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ITDK</td>
<td>26</td>
<td>2</td>
<td>15</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2.702</td>
</tr>
<tr>
<td>GNUTELLA-06</td>
<td>10</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>3.545</td>
</tr>
<tr>
<td>GNUTELLA-24</td>
<td>11</td>
<td>1</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>4.475</td>
</tr>
<tr>
<td>GNUTELLA-30</td>
<td>11</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>4.034</td>
</tr>
<tr>
<td>GNUTELLA-31</td>
<td>11</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>4.251</td>
</tr>
<tr>
<td>WEB-STANFORD</td>
<td>164</td>
<td>0</td>
<td>82</td>
<td>0</td>
<td>0</td>
<td>28</td>
<td>0.006</td>
</tr>
<tr>
<td>WEB-NOTREDAM</td>
<td>46</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.935</td>
</tr>
<tr>
<td>WEB-BERKSTAN</td>
<td>208</td>
<td>0</td>
<td>104</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>0.002</td>
</tr>
<tr>
<td>AMAZON-1</td>
<td>47</td>
<td>1</td>
<td>24</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>0.199</td>
</tr>
<tr>
<td>AMAZON-2</td>
<td>20</td>
<td>2</td>
<td>11</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>2.03</td>
</tr>
<tr>
<td>ROAD-EURO</td>
<td>62</td>
<td>0</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0.135</td>
</tr>
<tr>
<td>OPENFLIGHT</td>
<td>13</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.641</td>
</tr>
<tr>
<td>POWER-GRID</td>
<td>46</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1.409</td>
</tr>
<tr>
<td>ROAD-PA</td>
<td>772</td>
<td>32</td>
<td>417</td>
<td>21</td>
<td>22</td>
<td>80</td>
<td>22.545</td>
</tr>
</tbody>
</table>

Table 3. Qualities of a vertex $v$ most distant from a random vertex $u$, of a vertex $w$ of a $(u, v)$-geodesic at distance $rad(G)$ from $v$, and of a $BFS(w)$-tree $T_2$ rooted at vertex $w$. For each vertex $x \in V$, $k(x) := ecc_{T_2}(x) - ecc_G(x)$. Also, $k_{max} := \max_{x \in V} k(x)$ and $k_{avg} := \frac{1}{V} \sum_{x \in V} k(x)$.

For every graph $G = (V, E)$ in our dataset, we constructed also an arbitrary $BFS(w)$-tree $T_2 = (V, E')$, rooted at vertex $w$, and analyzed how well $T_2$ preserves or approximates the eccentricities of vertices in $G$. The value of $k_{max}$ is at most five for 23 graphs. The average distortion $k_{avg}$ is much smaller than $k_{max}$ in all graphs. The distributions of the values of $k(x)$ for all graphs are presented in Table 4 (see Appendix).

In Table 4 we compare these two eccentricity approximating spanning trees $T_1$ and $T_2$ with each other and with a third $BFS(c^*)$-tree $T_3$ which we have constructed starting from a randomly chosen central vertex $c^* \in C(G)$.

For each graph in the dataset, three values of $k_{max}$ ($k_{max}^{T_1}$, $k_{max}^{T_2}$ and $k_{max}^{T_3}$) and three values of $k_{avg}$ ($k_{avg}^{T_1}$, $k_{avg}^{T_2}$ and $k_{avg}^{T_3}$) are listed. We observe that the smallest $k_{max}$ (out of three) is achieved by tree $T_3$ in 28 graphs, by tree $T_2$ in 20 graphs and by tree $T_1$ in 20 graphs (in 14 graphs, the smallest $k_{max}$ is achieved by all three trees). The difference between the largest and the smallest $k_{max}$ of a graph is at most one for 26 graphs in the dataset. The largest difference is observed for ROAD-PA network: the largest $k_{max}$ (98) is given by tree $T_1$, the smallest $k_{max}$ (46) is given by tree $T_3$. Two other graphs have the difference larger than three: for SC-PPI network, the largest $k_{max}$ (7) is given by tree $T_2$, the smallest $k_{max}$ (3) is given by tree $T_1$; for POWER-GRID network, the largest $k_{max}$ (13) is given by tree $T_1$, the smallest $k_{max}$ (4) is shared by remaining trees $T_2$, $T_3$. Overall, we conclude that $k_{max}$ values for trees $T_1$ and $T_2$ are comparable and generally can be slightly worse than those for tree $T_3$. Similar observations hold also for the average distortion $k_{avg}$. Note, however, that for construction of trees $T_2$ and $T_3$ one needs to know $rad(G)$ or a central vertex of $G$, which are unlikely to be computable in subquadratic time (see Statement 2).
Table 4. Comparison of three BFS-trees $T_1$, $T_2$ and $T_3$. $T_3$ is a $BFS(c^*)$-tree rooted at a randomly picked central vertex $c^* \in C(G)$.

5.2 Estimation of Distances

Following Theorem 3, we experimented also on how well our approach approximates the distances in graphs from our dataset. To analyze the quality of approximation provided by our method for a given graph $G = (V, E)$, for every $\delta := 0, 1, 2, \ldots$, we computed an estimate $\hat{d}_\delta(x, y)$ on $d_G(x, y)$ and the error $\Delta_{xy}(\delta) = \hat{d}_\delta(x, y) - d_G(x, y)$ for all $x, y \in V$. In Table 5, we report $\Delta_{max}(\delta) = \max_{x, y \in V} \Delta_{xy}(\delta)$ and $\Delta_{avg}(\delta) = \sum_{x, y \in V} \Delta_{xy}(\delta)$ for the smallest $\delta$ such that $\Delta_{max}(\delta) \leq \delta + 1$. We omitted some very large graphs in this experiment. For some other large graphs, we did only sampling; we calculated $\Delta_{max}(\delta)$ and $\Delta_{avg}(\delta)$ based only on a set of sampled vertices. We sampled vertices that are most distant from the root. The number of sampled vertices ranged from 10 to 100 in each network. For all networks investigated, the average error $\Delta_{avg}(\delta)$ was very small, less than 1 even for infrastructure networks. That is, the maximum error $\Delta_{max}(\delta)$ was realized on a very small number of vertex pairs. The maximum error $\Delta_{max}(\delta)$ was 2 for three networks, was 3 for five networks, was 4 for ten networks (including infrastructure network OPENFLIGHT), and was at most 6 for all except one social network DUTCH-ELITE and two infrastructure networks: ROAD-EURO and POWER-GRID. The largest $\Delta_{max}(\delta)$ value had expectedly POWER-GRID network whose hyperbolicity is 10.

Acknowledgements

The research of V.C., M.H., and Y.V. was supported by ANR project DISTANCIA (ANR-17-CE40-0015).

References

1. https://snap.stanford.edu/data/
Table 5. Distance approximations: for every \(x, y \in V\), \(\Delta_{xy}(\delta) = \hat{d}_\delta(x, y) - d_G(x, y)\); \(\Delta_{\max}(\delta) = \max_{x,y \in V} \Delta_{xy}(\delta)\); \(\Delta_{\text{avg}}(\delta) = \frac{1}{n^2} \sum_{x,y \in V} \Delta_{xy}(\delta)\); \(\delta\) is defined as the smallest \(0 \leq \delta \leq \text{diam}(G)\) such that \(\Delta_{\max}(\delta) \leq \delta + 1\). Due to large sizes of some networks, the values of \(\Delta_{\max}(\delta)\) and \(\Delta_{\text{avg}}(\delta)\) for networks marked with * were computed only for some sampled vertices (we sampled vertices that are most distant from the root). The number of sampled vertices ranged from 10 to 100 in each network.

<table>
<thead>
<tr>
<th>Network</th>
<th>(\text{diam}(G))</th>
<th>(\text{rad}(G))</th>
<th>(\delta)</th>
<th>(\Delta_{\max}(\delta))</th>
<th>(\Delta_{\text{avg}}(\delta))</th>
<th>(\text{ecc}(s))</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUTCH-ELITE</td>
<td>22</td>
<td>12</td>
<td>8</td>
<td>8</td>
<td>0.177</td>
<td>16</td>
</tr>
<tr>
<td>FACEBOOK</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0.169</td>
<td>6</td>
</tr>
<tr>
<td>EVA</td>
<td>18</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>0.044</td>
<td>12</td>
</tr>
<tr>
<td>SLASHDOT*</td>
<td>12</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>0.028</td>
<td>8</td>
</tr>
<tr>
<td>LOANS*</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>0.213</td>
<td>6</td>
</tr>
<tr>
<td>TWITTER*</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>0.156</td>
<td>6</td>
</tr>
<tr>
<td>EMAIL-VIRGILI</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>0.39</td>
<td>6</td>
</tr>
<tr>
<td>EMAIL-ENRON</td>
<td>13</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>0.06</td>
<td>9</td>
</tr>
<tr>
<td>EMAIL-EU*</td>
<td>14</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>0.005</td>
<td>10</td>
</tr>
<tr>
<td>CE-METABOLIC</td>
<td>7</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0.125</td>
<td>4</td>
</tr>
<tr>
<td>SC-PP1</td>
<td>19</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>0.19</td>
<td>13</td>
</tr>
<tr>
<td>YEAST-PP1</td>
<td>11</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>0.239</td>
<td>8</td>
</tr>
<tr>
<td>HOMO-PI</td>
<td>10</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>0.02</td>
<td>7</td>
</tr>
<tr>
<td>AS-GRAPH-1</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>0.061</td>
<td>8</td>
</tr>
<tr>
<td>AS-GRAPH-2</td>
<td>11</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>0.034</td>
<td>8</td>
</tr>
<tr>
<td>ROUTVIEW</td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>0.035</td>
<td>9</td>
</tr>
<tr>
<td>AS-CAIDA</td>
<td>17</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>0.022</td>
<td>14</td>
</tr>
<tr>
<td>ITDK*</td>
<td>26</td>
<td>14</td>
<td>5</td>
<td>4</td>
<td>0.15</td>
<td>19</td>
</tr>
<tr>
<td>GNUTELLA-06</td>
<td>9</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>0.331</td>
<td>8</td>
</tr>
<tr>
<td>GNUTELLA-24</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0.128</td>
<td>9</td>
</tr>
<tr>
<td>GNUTELLA-30*</td>
<td>11</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>0.439</td>
<td>8</td>
</tr>
<tr>
<td>GNUTELLA-31*</td>
<td>11</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>0.386</td>
<td>9</td>
</tr>
<tr>
<td>ROAD-EURO</td>
<td>62</td>
<td>31</td>
<td>21</td>
<td>11</td>
<td>0.927</td>
<td>39</td>
</tr>
<tr>
<td>OPENFLIGHT</td>
<td>13</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>0.029</td>
<td>10</td>
</tr>
<tr>
<td>POWER-GRID</td>
<td>46</td>
<td>23</td>
<td>17</td>
<td>17</td>
<td>0.518</td>
<td>38</td>
</tr>
</tbody>
</table>

2. http://konect.uni-koblenz.de/networks/
5. Center for applied Internet data analysis, http://www.caida.org/data/as-relationships
7. University of oregon route-views project http://www.routeviews.org/


41. N. Cohen, D. Coudert, and A. Lancin, Exact and approximate algorithms for computing the hyperbolicity of large-scale graphs, RR-8074 (hal-00735481v3), INRIA, 2012.


52. F.F. Dragan, M. Habib, L. Viennot, Revisiting Radius, Diameter, and all Eccentricity Computation in Graphs through Certificates, CoRR abs/1803.04660 (2018)
71. J. Kunegis, Prosper loans, KONECT, the Koblenz Network Collection, 2016.
91. V. Vassilevska Williams, Hardness of easy problems: basing hardness on popular conjectures such as the strong exponential time hypothesis, *IPEC 2015*, pp. 17–29.
## Appendix

<table>
<thead>
<tr>
<th>Network</th>
<th>$k_{max}$</th>
<th>$k_{avg}$</th>
<th>$%$ of vertices with $k(x) = 0$</th>
<th>$%$ of vertices with $k(x) = 1$</th>
<th>$%$ of vertices with $k(x) = 2$</th>
<th>$%$ of vertices with $k(x) = 3$</th>
<th>$%$ of vertices with $k(x) = 4$</th>
<th>$%$ of vertices with $k(x) \geq 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUTCH-ELITE</td>
<td>6</td>
<td>2.35</td>
<td>14.9</td>
<td>0</td>
<td>54.3</td>
<td>0</td>
<td>29.1</td>
<td>0</td>
</tr>
<tr>
<td>FACEBOOK</td>
<td>2</td>
<td>0.686</td>
<td>51.9</td>
<td>27.6</td>
<td>20.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVA</td>
<td>2</td>
<td>0.571</td>
<td>47.6</td>
<td>47.7</td>
<td>4.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLASHDOT</td>
<td>3</td>
<td>1.777</td>
<td>2.3</td>
<td>24.1</td>
<td>67.1</td>
<td>6.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOANS</td>
<td>3</td>
<td>2.06</td>
<td>0.1</td>
<td>13.9</td>
<td>66.3</td>
<td>19.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TWITTER</td>
<td>4</td>
<td>2.565</td>
<td>0.1</td>
<td>$\approx 1$</td>
<td>44.4</td>
<td>51.2</td>
<td>3.4</td>
<td></td>
</tr>
<tr>
<td>EMAIL-VIRGILI</td>
<td>4</td>
<td>2.729</td>
<td>0.1</td>
<td>2.3</td>
<td>32</td>
<td>55.7</td>
<td>9.9</td>
<td></td>
</tr>
<tr>
<td>EMAIL-ENRON</td>
<td>2</td>
<td>0.906</td>
<td>23.4</td>
<td>62.6</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMAIL-EU</td>
<td>2</td>
<td>0.002</td>
<td>99.8</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WIKITALK-CHINA</td>
<td>3</td>
<td>2.076</td>
<td>$\approx 0$</td>
<td>0.01</td>
<td>92.4</td>
<td>7.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CE-METABOLIC</td>
<td>3</td>
<td>1.982</td>
<td>0.2</td>
<td>7.5</td>
<td>86.1</td>
<td>6.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC-PP1</td>
<td>3</td>
<td>0.981</td>
<td>32.4</td>
<td>41.6</td>
<td>21.5</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YEAST-PP1</td>
<td>3</td>
<td>1.872</td>
<td>2</td>
<td>25.4</td>
<td>55.8</td>
<td>16.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOMO-PP</td>
<td>2</td>
<td>0.747</td>
<td>34.2</td>
<td>56.9</td>
<td>8.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS-GRAF1-1</td>
<td>3</td>
<td>1.791</td>
<td>0.5</td>
<td>24.9</td>
<td>69.7</td>
<td>4.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS-GRAF1-2</td>
<td>3</td>
<td>1.124</td>
<td>9.6</td>
<td>68.5</td>
<td>21.7</td>
<td>0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS-GRAF1-3</td>
<td>2</td>
<td>0.828</td>
<td>27.8</td>
<td>61.6</td>
<td>10.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROUTEVIEW</td>
<td>2</td>
<td>0.329</td>
<td>69.7</td>
<td>27.6</td>
<td>2.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS-Caida</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITOK</td>
<td>4</td>
<td>2.108</td>
<td>0.3</td>
<td>12</td>
<td>64.5</td>
<td>22.8</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>GNUTELLA-06</td>
<td>4</td>
<td>2.567</td>
<td>0.3</td>
<td>5.7</td>
<td>41.1</td>
<td>48.8</td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>GNUTELLA-24</td>
<td>5</td>
<td>2.097</td>
<td>0.2</td>
<td>1.5</td>
<td>37</td>
<td>50.7</td>
<td>10.5</td>
<td>0.1</td>
</tr>
<tr>
<td>GNUTELLA-30</td>
<td>5</td>
<td>3.167</td>
<td>0.1</td>
<td>1.8</td>
<td>13</td>
<td>52.4</td>
<td>31.8</td>
<td>0.9</td>
</tr>
<tr>
<td>GNUTELLA-31</td>
<td>6</td>
<td>4.176</td>
<td>0.01</td>
<td>0.2</td>
<td>1.4</td>
<td>13.3</td>
<td>51.1</td>
<td>33.4</td>
</tr>
<tr>
<td>WEB-STANFORD</td>
<td>28</td>
<td>0.006</td>
<td>99.9</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
</tr>
<tr>
<td>WEB-NOTREDAM</td>
<td>2</td>
<td>0.935</td>
<td>7.1</td>
<td>92.4</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEB-BERKSTAN</td>
<td>22</td>
<td>0.002</td>
<td>99.9</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
</tr>
<tr>
<td>AMAZON-1</td>
<td>6</td>
<td>0.991</td>
<td>28.1</td>
<td>48</td>
<td>21.5</td>
<td>1.5</td>
<td>0.5</td>
<td>0.3</td>
</tr>
<tr>
<td>AMAZON-2</td>
<td>6</td>
<td>3.735</td>
<td>0.1</td>
<td>0.3</td>
<td>3.6</td>
<td>33.9</td>
<td>46.5</td>
<td>15.3</td>
</tr>
<tr>
<td>ROAD-EUROR</td>
<td>8</td>
<td>0.135</td>
<td>97.4</td>
<td>0.3</td>
<td>0.1</td>
<td>0.4</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>OPENFLIGHT</td>
<td>3</td>
<td>1.879</td>
<td>0.2</td>
<td>23.9</td>
<td>63.7</td>
<td>12.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POWER-GRID</td>
<td>13</td>
<td>5.735</td>
<td>14.3</td>
<td>13.1</td>
<td>1.6</td>
<td>1.6</td>
<td>3.9</td>
<td>8.7</td>
</tr>
<tr>
<td>ROAD-PA</td>
<td>98</td>
<td>23.339</td>
<td>0.02</td>
<td>1.5</td>
<td>0.1</td>
<td>2.9</td>
<td>0.2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Table 6. Distribution of values $k(x) = ecc_{T_1}(x) - ecc_G(x),$ $x \in V$. $k_{max} := \max_{x \in V} k(x),$ $k_{avg} := \frac{1}{n} \sum_{x \in V} k(x).$
Table 7. Distribution of values $k(x) = e_{cc_T}(x) - e_{cc}(x)$, $x \in V$. $k_{max} := \max_{x \in V} k(x)$. $k_{avg} := \frac{1}{n} \sum_{x \in V} k(x)$.

<table>
<thead>
<tr>
<th>Network</th>
<th>$k_{max}$</th>
<th>$k_{avg}$</th>
<th>% of vertices with $k(x) = 0$</th>
<th>% of vertices with $k(x) = 1$</th>
<th>% of vertices with $k(x) = 2$</th>
<th>% of vertices with $k(x) = 3$</th>
<th>% of vertices with $k(x) = 4$</th>
<th>% of vertices with $k(x) \geq 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUTCH-ELITE</td>
<td>6</td>
<td>2.431</td>
<td>16.1</td>
<td>0</td>
<td>47.1</td>
<td>0</td>
<td>35.9</td>
<td>0</td>
</tr>
<tr>
<td>FACEBOOK</td>
<td>3</td>
<td>0.704</td>
<td>43.6</td>
<td>42.5</td>
<td>13.8</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVA</td>
<td>2</td>
<td>0.572</td>
<td>47.6</td>
<td>47.6</td>
<td>4.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLASHDOT</td>
<td>3</td>
<td>1.88</td>
<td>0.1</td>
<td>17.7</td>
<td>76.2</td>
<td>$\approx 0$</td>
<td></td>
<td>$\approx 0$</td>
</tr>
<tr>
<td>LOANS</td>
<td>3</td>
<td>2.031</td>
<td>0.1</td>
<td>14</td>
<td>68.7</td>
<td>17.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TWITTER</td>
<td>3</td>
<td>1.821</td>
<td>3.1</td>
<td>$\approx 20$</td>
<td>68.6</td>
<td>8.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMAIL-VIRGILI</td>
<td>4</td>
<td>1.932</td>
<td>4.3</td>
<td>22.8</td>
<td>48.4</td>
<td>24.1</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>EMAIL-ENRON</td>
<td>2</td>
<td>0.903</td>
<td>22.4</td>
<td>64.8</td>
<td>12.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMAIL-EU</td>
<td>2</td>
<td>0.002</td>
<td>99.9</td>
<td>0.03</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WIKITALK-CHINA</td>
<td>3</td>
<td>1.791</td>
<td>$\approx 0$</td>
<td>21</td>
<td>79</td>
<td>$0.008$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CE-METABOLIC</td>
<td>1</td>
<td>0.349</td>
<td>65.1</td>
<td>34.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC-PPPI</td>
<td>7</td>
<td>4.196</td>
<td>1.3</td>
<td>4.1</td>
<td>6.2</td>
<td>13.4</td>
<td>27.2</td>
<td>35.9</td>
</tr>
<tr>
<td>YEAST-PPPI</td>
<td>4</td>
<td>2.558</td>
<td>0.7</td>
<td>5.9</td>
<td>$\approx 36$</td>
<td>51.7</td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td>HOMO-PI</td>
<td>2</td>
<td>0.612</td>
<td>41.6</td>
<td>55.5</td>
<td>2.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS-GRAPH-1</td>
<td>2</td>
<td>0.887</td>
<td>19.6</td>
<td>72.2</td>
<td>8.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS-GRAPH-2</td>
<td>2</td>
<td>0.834</td>
<td>25.7</td>
<td>65.3</td>
<td>$\approx 9$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS-GRAPH-3</td>
<td>2</td>
<td>0.312</td>
<td>70.4</td>
<td>28</td>
<td>1.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROUTEVIEW</td>
<td>2</td>
<td>0.329</td>
<td>69.7</td>
<td>27.6</td>
<td>2.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS-CAIDA</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITDK</td>
<td>5</td>
<td>2.702</td>
<td>0.3</td>
<td>3.4</td>
<td>28.6</td>
<td>61.4</td>
<td>6.3</td>
<td>$\approx 0$</td>
</tr>
<tr>
<td>GNUTELLA-06</td>
<td>5</td>
<td>3.543</td>
<td>0.01</td>
<td>0.7</td>
<td>5.9</td>
<td>37.2</td>
<td>50.9</td>
<td>5.3</td>
</tr>
<tr>
<td>GNUTELLA-24</td>
<td>6</td>
<td>4.475</td>
<td>0.02</td>
<td>0.1</td>
<td>0.7</td>
<td>8.6</td>
<td>38.3</td>
<td>46.5</td>
</tr>
<tr>
<td>GNUTELLA-30</td>
<td>6</td>
<td>4.034</td>
<td>0.02</td>
<td>0.2</td>
<td>2.6</td>
<td>16.4</td>
<td>54.8</td>
<td>25.1</td>
</tr>
<tr>
<td>GNUTELLA-31</td>
<td>6</td>
<td>4.251</td>
<td>0.01</td>
<td>0.1</td>
<td>1.3</td>
<td>11.6</td>
<td>48.4</td>
<td>37.8</td>
</tr>
<tr>
<td>WEB-STANFORD</td>
<td>28</td>
<td>0.086</td>
<td>99.9</td>
<td>$\approx 0$</td>
<td>$\approx 0$</td>
<td>0.04</td>
<td>$\approx 0$</td>
<td>0.02</td>
</tr>
<tr>
<td>WEB-NOTREDAM</td>
<td>2</td>
<td>0.935</td>
<td>7.1</td>
<td>92.3</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEB-BERKSTAN</td>
<td>22</td>
<td>0.002</td>
<td>99.97</td>
<td>$\approx 0$</td>
<td>0</td>
<td>0.02</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>AMAZON-1</td>
<td>7</td>
<td>0.919</td>
<td>49.7</td>
<td>21.7</td>
<td>18.6</td>
<td>8.1</td>
<td>1.1</td>
<td>0.4</td>
</tr>
<tr>
<td>AMAZON-2</td>
<td>5</td>
<td>2.03</td>
<td>1.2</td>
<td>15.1</td>
<td>65</td>
<td>17.1</td>
<td>1.6</td>
<td>$\approx 0$</td>
</tr>
<tr>
<td>ROAD-EURO</td>
<td>8</td>
<td>0.135</td>
<td>97.4</td>
<td>0.3</td>
<td>0.1</td>
<td>0.4</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>OPENFLIGHT</td>
<td>2</td>
<td>0.641</td>
<td>36.1</td>
<td>63.7</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POWER-GRID</td>
<td>4</td>
<td>1.409</td>
<td>46.3</td>
<td>13.1</td>
<td>12.6</td>
<td>9.1</td>
<td>18.8</td>
<td></td>
</tr>
<tr>
<td>ROAD-PA</td>
<td>80</td>
<td>22.545</td>
<td>0.7</td>
<td>20.9</td>
<td>0.3</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
</tr>
</tbody>
</table>