



Fast approximation of eccentricities and distances in hyperbolic graphs

*Victor Chepoi*¹ *Feodor F. Dragan*² *Michel Habib*³ *Yann Vaxès*¹
*Hend Alrasheed*⁴

¹Laboratoire d'Informatique et Systèmes, Aix-Marseille Univ, CNRS, and Univ. de Toulon Faculté des Sciences de Luminy, F-13288 Marseille Cedex 9, France

²Algorithmic Research Laboratory, Department of Computer Science, Kent State University, Kent, Ohio, USA

³Institut de Recherche en Informatique Fondamentale, University Paris Diderot - Paris7, F-75205 Paris Cedex 13, France

⁴Information Technology Department, King Saud University, Riyadh, Saudi Arabia

Submitted: October 2018	Reviewed: March 2019	Revised: April 2019	Accepted: May 2019	Final: May 2019
		Published: June 2019		
	Article type: Regular paper		Communicated by: G. Liotta	

Results of this paper are partially presented at the COCOA 2018 conference [37]. The research of V.C., M.H., and Y.V. was supported by ANR project DISTANCIA (ANR-17-CE40-0015).
E-mail addresses: victor.chepoi@lif.univ-mrs.fr (Victor Chepoi) dragan@cs.kent.edu (Feodor F. Dragan) habib@liafa.univ-paris-diderot.fr (Michel Habib) yann.vaxes@lif.univ-mrs.fr (Yann Vaxès) halrasheed@ksu.edu.sa (Hend Alrasheed)

Abstract

We show that the eccentricities 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 $\text{ecc}_G(v) := \max\{d_G(u, v) : u \in V\}$ such that $\text{ecc}_G(v) \leq \hat{e}(v) \leq \text{ecc}_G(v) + c\delta$ for a small constant c . We prove that every δ -hyperbolic graph G has a shortest path tree T , constructible in linear time, such that for every vertex v of G , $\text{ecc}_G(v) \leq \text{ecc}_T(v) \leq \text{ecc}_G(v) + c\delta$, where $\text{ecc}_T(v) := \max\{d_T(u, v) : u \in V\}$. 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 eccentricities 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

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 $\text{ecc}_G(v)$, is the largest distance from v to any other vertex, i.e., $\text{ecc}_G(v) = \max_{u \in V} d_G(v, u)$. The *radius* $\text{rad}(G)$ of a graph G is the minimum eccentricity of a vertex in G , i.e., $\text{rad}(G) = \min_{v \in V} \text{ecc}_G(v)$. The *diameter* $\text{diam}(G)$ of a graph G is the maximum eccentricity of a vertex in G , i.e., $\text{diam}(G) = \max_{v \in V} \text{ecc}_G(v)$. The *center* $C(G) = \{c \in V : \text{ecc}_G(c) = \text{rad}(G)\}$ of a graph G is the set of vertices with minimum eccentricity.

The diameter $\text{diam}(G)$ and the radius $\text{rad}(G)$ 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* $\text{ecc}(v)$ of a vertex v is used to measure the

importance of v in the network: the (*eccentricity*) *centrality index* of v [68] is defined as $\frac{1}{\text{ecc}(v)}$.

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 [9, 18, 25, 29, 33, 38, 43, 53, 56, 62, 77]. 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 [9, 21, 80], 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 [8] 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., [81, 92]).

Motivated by these negative results, researches 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 $o(nm)$ 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 $\tilde{O}(mn^{2/3})^1$ time was recently obtained in [32] (see also [12] for an earlier $\tilde{O}(n^2 + m\sqrt{n})$ time algorithm and [80] 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 $O(n^{2-\epsilon})$ time algorithm can exist that $(3/2 - \epsilon')$ -approximates (for $\epsilon, \epsilon' > 0$) the diameter [80] and the radius [9] 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 $\tilde{O}(m^{3/2})$ time [32] and, under plausible assumptions, no $O(n^{2-\epsilon})$ time algorithm can exist that $(5/3 - \epsilon')$ -approximates (for $\epsilon, \epsilon' > 0$) the eccentricities of all vertices in sparse graphs [9]. Better approximation algorithms are known for some special classes of graphs [26, 34, 35, 43, 44, 50, 52, 54, 89]. A number of heuristics for approximating diameters, radii and eccentricities in real-world graphs were proposed and investigated in [10, 21, 22, 23, 68, 30, 48].

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 $\tilde{O}(\min\{n^{3/2}m^{1/2}, n^{7/3}\})$

¹ \tilde{O} hides a polylog factor.

time and hence improves the runtime of an earlier algorithm from [12]. In [19], an $\tilde{O}(n^2)$ 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 $O(n^{2.24+o(1)}\epsilon^{-3}\log(n/\epsilon))$ time algorithm that computes an approximation of all distances with a multiplicative error of $(1+\epsilon)$ 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, 51, 86] 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 folklore 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. One 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 $d_G(y, z)$ is not the diameter but is still its good approximation: $d_G(y, z) \geq \text{diam}(G) - 2$. Even better, the diameter of any chordal graph can be approximated in linear time with an additive error 1 [54]. 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 $\text{diam}(G)$, $\text{rad}(G)$, $C(G)$, $\text{ecc}_G(v)$ 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 δ -hyperbolic [14, 59, 27, 60] if for any four vertices w, v, x, y of G , the two largest of the three distance sums $d_G(w, v) + d_G(x, y)$, $d_G(w, x) + d_G(v, y)$, $d_G(w, y) + d_G(v, x)$ differ by at most $2\delta \geq 0$. The *hyperbolicity* $\delta(G)$ of a graph G is the smallest number δ such that G is δ -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 net-

works, biological networks, and others) are tree-like from a metric point of view [10, 11, 20] or have small hyperbolicity [67, 76, 82]. It has been suggested in [76], 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 also many algorithmic advantages. Efficient approximate solutions are attainable for a number of optimization problems [35, 36, 39, 40, 45, 57, 87].

In [35] we initiated the investigation of diameter, center, and radius problems for δ -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 δ -hyperbolic graph and $\{y, z\}$ is the pair returned after two FP scans, then $d_G(y, z) \geq \text{diam}(G) - 2\delta$, $\text{diam}(G) \geq 2\text{rad}(G) - 4\delta - 1$, $\text{diam}(C(G)) \leq 4\delta + 1$, 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 δ -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* ($\text{diam}(G)$ 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 δ , $\text{ecc}_G(v)$ is equal to $d_G(v, C(G))$ plus $\text{rad}(G)$. 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 [79]. 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 [79]. 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 $k = 2$ [69, 73, 78], in asteroidal-triple-free graph [69], strongly chordal graphs [24] and dually chordal graphs [24] for $k = 3$. On the other hand, although for every k there is a chordal graph without an additive tree k -spanner [69, 78], yet as Prisner demonstrated in [79], every chordal graph has an eccentricity 2-approximating spanning tree.

Later this result was extended in [52] 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 [79] to all δ -hyperbolic graphs.

As our second main result, we show that in every δ -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 [31], this demonstrates an equivalence between approximating the hyperbolicity and approximating the distances in graphs. Note that every δ -hyperbolic graph G admits a distance approximating tree T [35, 36], 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 δ . 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 [69, 78].

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 Additional notions and notations

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 *kth-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., $ecc_G(x) = ecc_G(y) = d_G(x, y)$.

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 δ -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 δ -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 δ -hyperbolic (or of hyperbolicity δ) if the metric space (V, d_G) is δ -hyperbolic, where d_G is the standard shortest path metric defined on G .

δ -Hyperbolic graphs generalize k -chordal graphs and graphs of bounded tree-length: each k -chordal graph has the tree-length at most $\lfloor \frac{k}{2} \rfloor$ [47] and each tree-length λ graph has hyperbolicity at most λ [35]. Recall that a graph is k -chordal if its induced cycles are of length at most k , and it is of tree-length λ if it has a Robertson-Seymour tree-decomposition into bags of diameter at most λ [47].

For geodesic metric spaces and graphs there exist several equivalent definitions of δ -hyperbolicity involving different but comparable values of δ [14, 27, 59, 60]. 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 $\alpha_y := (x|z)_y = (d(y, x) + d(y, z) - d(x, z))/2$ 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 φ which maps $\Delta(x, y, z)$ to a tripod $T(x, y, z)$ consisting of three solid segments $[x, m], [y, m]$, and $[z, m]$ of lengths α_x, α_y , and α_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 δ -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 δ -thin is called a graph with δ -thin triangles, and δ 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.

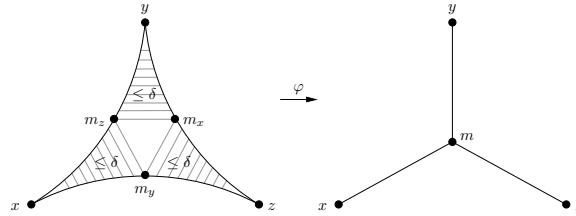


Figure 1: A geodesic triangle $\Delta(x, y, z)$, the points m_x, m_y, m_z , and the tripod $T(x, y, z)$

Proposition 1 ([14, 27, 59, 60]) *Geodesic triangles of geodesic δ -hyperbolic spaces or graphs are 4δ -thin. Conversely, geodesic spaces or graphs with δ -thin triangles are δ -hyperbolic.*

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 δ -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. [80] 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 [90] 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., [9, 21, 91] and papers cited therein).

Since all geodesic triangles of a graph constructed in the reduction in [80] are 2-thin, we can rephrase the result from [80] as follows.

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. [9] 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. [9] 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 [9] are 2-thin, rephrasing that result from [9], 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 δ -thin triangles the eccentricities of all vertices can be computed in total linear time with an additive error depending on δ . 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 δ . For these purposes, we revisit and extend several results from our previous paper [35] concerning the linear time approximation of diameter, radius, and centers of δ -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 $k = 0, \dots, \text{diam}(G) - \text{rad}(G)$ 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 δ -thin triangles, a middle vertex c of any geodesic between two mutually distant vertices has the eccentricity close to $\text{rad}(G)$ ($\text{ecc}_G(c) \leq \text{rad}(G) + \delta$) and is not too far from the center $C(G)$ of G (all central vertices of G are within distance at most $2\delta + 1$ from c). See Fig. 2 for an illustration. Furthermore, the distance between any mutually distant vertices u and v (and, hence, the diameter of G) is at least $2\text{rad}(G) - 2\delta - 1$.

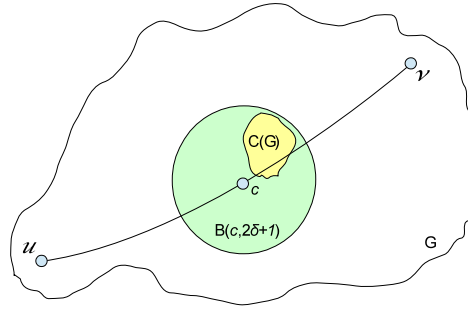


Figure 2: Illustration to Proposition 2.

Proposition 2 Let G be a graph with δ -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 $\text{ecc}_G(c) \leq \lceil \frac{d_G(u, v)}{2} \rceil + \delta \leq \text{rad}(G) + \delta$.
- (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 $(x|u)_v = \frac{1}{2}(d_G(x, v) + d_G(v, u) - d_G(x, u))$ from v and hence at distance $(x|v)_u = \frac{1}{2}(d_G(x, u) + d_G(v, u) - d_G(x, v))$ from u (see Fig. 3). 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., $d_G(v, c^*) \leq d_G(v, m_x) = (x|u)_v$, and hence $(x|v)_u \leq (x|u)_v$. Since $d_G(v, x) \leq d_G(v, u)$, we also get $(u|v)_x \leq (x|v)_u$.

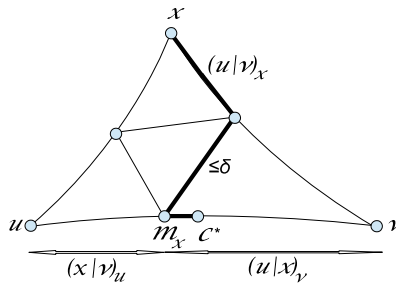


Figure 3: Illustration to the proof of Proposition 2(a).

(a) By the triangle inequality and since $d_G(u, v) \leq \text{diam}(G) \leq 2\text{rad}(G)$, we get (see Fig. 3)

$$\begin{aligned} d_G(x, c^*) &\leq (u|v)_x + \delta + d_G(u, c^*) - (x|v)_u \\ &\leq d_G(u, c^*) + \delta = \frac{d_G(u, v)}{2} + \delta \leq \text{rad}(G) + \delta. \end{aligned}$$

(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 $\text{ecc}_G(c) \leq \text{ecc}_G(c^*) + \frac{1}{2}$. 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

$$\frac{d_G(u, v)}{2} + \frac{1}{2} + \delta = \lceil \frac{d_G(u, v)}{2} \rceil + \delta \leq \lceil \frac{2\text{rad}(G) - 1}{2} \rceil + \delta = \text{rad}(G) + \delta.$$

(c) Since a middle vertex c of any (u, v) -geodesic sees all vertices of G within distance at most $\lceil \frac{d_G(u, v)}{2} \rceil + \delta$, if $d_G(u, v) \leq 2\text{rad}(G) - 2\delta - 2$, then

$$\text{ecc}_G(c) \leq \lceil \frac{d_G(u, v)}{2} \rceil + \delta \leq \lceil \frac{2\text{rad}(G) - 2\delta - 2}{2} \rceil + \delta < \text{rad}(G),$$

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

$$\begin{aligned} d_G(x, c^*) &\leq (u|v)_x + \delta + (x|u)_v - d_G(v, c^*) = d_G(v, x) - d_G(v, c^*) + \delta \\ &\leq \text{rad}(G) + k - \frac{d_G(u, v)}{2} + \delta \leq k + 2\delta + \frac{1}{2}. \end{aligned}$$

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

$$\begin{aligned} d_G(x, c) &\geq \text{ecc}_G(x) - \text{ecc}_G(c) = k + \text{rad}(G) - \text{ecc}_G(c) \\ &\geq (k + \text{rad}(G)) - (\text{rad}(G) + \delta) = k - \delta. \end{aligned}$$

□

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 $d_G(x, C(G))$ plus $\text{rad}(G)$.

Corollary 1 *For every vertex x of a graph G with δ -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), $d_G(c, c_x) \leq 2\delta + 1$ and $d_G(x, c) \leq k + 2\delta + 1 = \text{ecc}_G(x) - \text{rad}(G) + 2\delta + 1$. Hence,

$$d_G(x, C(G)) = d_G(x, c_x) \leq d_G(x, c) + d_G(c, c_x) \leq d_G(x, c) + 2\delta + 1$$

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 $\text{ecc}_G(x) = d_G(x, C(G)) + \text{rad}(G)$ holds for every vertex of a graph G if and only if the eccentricity function $\text{ecc}_G(\cdot)$ on G is unimodal (that is, every local minimum is a global minimum)[49]. In particular, for every tree T and every its vertex x , $\text{ecc}_T(x) = d_T(x, C(T)) + \text{rad}(T)$ holds. A slightly weaker condition holds for all chordal graphs [52]: for every vertex x of a chordal graph G , $d_G(x, C(G)) + \text{rad}(G) - 1 \leq \text{ecc}_G(x) \leq d_G(x, C(G)) + \text{rad}(G)$.

As we show later, Proposition 2 can be used to compute in $O(\delta|E|)$ time sharp approximations of radius, diameter and all eccentricities in a graph G with δ -thin triangles. To get pure linear-time sharp approximations, we will need the following proposition. It states that the eccentricity of a vertex v which is most distant from an arbitrary vertex u is at least $\text{diam}(G) - 2\delta$. Furthermore, if t is a vertex most distant from v , then a vertex c of any (v, t) -geodesic that is at distance $\lceil \frac{d_G(v, t)}{2} \rceil$ from t has the eccentricity at most $\text{rad}(G) + 3\delta$ and contains the center $C(G)$ of G in $B(c, 3\delta + 1)$ (i.e., $C(G) \subseteq B(c, 3\delta + 1)$).

Proposition 3 *Let G be a graph with δ -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 $\text{rad}(G)$ from v , then $\text{ecc}_G(w) \leq \text{rad}(G) + \delta$.*
- (b) *For every pair of vertices $x, y \in V$, $\max\{d_G(v, x), d_G(v, y)\} \geq d_G(x, y) - 2\delta$.*
- (c) *$\text{ecc}_G(v) \geq \text{diam}(G) - 2\delta \geq 2\text{rad}(G) - 4\delta - 1$.*
- (d) *If $t \in F(v)$, c is a vertex of a (v, t) -geodesic at distance $\lceil \frac{d_G(v, t)}{2} \rceil$ from t and $x \in C^k(G)$, then $\text{ecc}_G(c) \leq \text{rad}(G) + 3\delta$ 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 $d_G(w, x) = ecc_G(w)$. 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 $(x|u)_v = \frac{1}{2}(d_G(x, v) + d_G(v, u) - d_G(x, u))$ from v and hence at distance $(x|v)_u = \frac{1}{2}(d_G(x, u) + d_G(v, u) - d_G(x, v))$ from u . We distinguish between two cases: w is between u and m_x or w is between v and m_x in $[u, v]$.

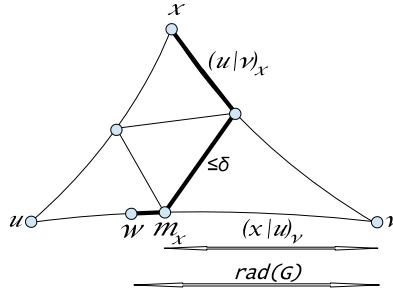


Figure 4: Illustration to the proof of Proposition 3(a).

In the first case, by the triangle inequality and $d_G(u, x) \leq d_G(u, v)$ (and hence, $(u|x)_v \geq (u|v)_x$), we get (see Fig. 4)

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

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

$$\begin{aligned} d_G(w, x) &\leq (u|x)_v - rad(G) + \delta + (u|v)_x \\ &\leq d_G(x, v) - rad(G) + \delta \\ &\leq 2rad(G) - rad(G) + \delta = rad(G) + \delta. \end{aligned}$$

(b) Consider an arbitrary (u, v) -geodesic $[u, v]$. 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 $\Delta(u, v, y) := [u, v] \cup [v, y] \cup [y, u]$ 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 $(x|u)_v = \frac{1}{2}(d_G(x, v) + d_G(v, u) - d_G(x, u))$ from v and hence at distance $(x|v)_u = \frac{1}{2}(d_G(x, u) + d_G(v, u) - d_G(x, v))$ from u . Let m_y be a point on $[u, v]$ which is at distance $(y|u)_v = \frac{1}{2}(d_G(y, v) + d_G(v, u) - d_G(y, u))$ from v and hence at distance $(y|v)_u = \frac{1}{2}(d_G(y, u) + d_G(v, u) - d_G(y, v))$ from u . Without loss of generality, assume that m_x is on $[u, v]$ between v and m_y (see Fig. 5).

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

$$\begin{aligned} d_G(x, y) &\leq (u|v)_x + \delta + ((y|u)_v - (u|x)_v) + \delta + (u|v)_y \\ &\leq (u|x)_v - (u|x)_v + 2\delta + (y|u)_v + (u|v)_y \\ &= d_G(v, y) + 2\delta. \end{aligned}$$

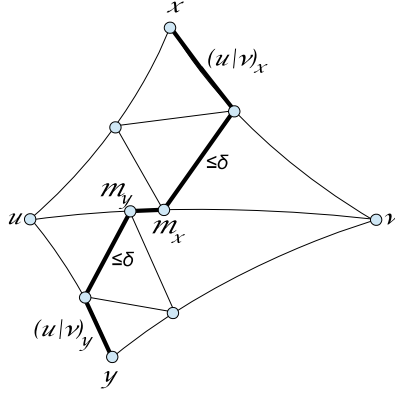


Figure 5: Illustration to the proof of Proposition 3(b).

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., $d_G(x, y) = \text{diam}(G)$, then, by (b) and Proposition 2(c),

$$\begin{aligned} \text{ecc}_G(v) &\geq \max\{d_G(v, x), d_G(v, y)\} \\ &\geq d_G(x, y) - 2\delta = \text{diam}(G) - 2\delta \\ &\geq 2\text{rad}(G) - 4\delta - 1. \end{aligned}$$

(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 $\text{rad}(G)$ from t , and c be a vertex of $[v, t]$ at distance $\lceil \frac{d_G(v, t)}{2} \rceil$ from t . We know by (a) that $\text{ecc}_G(w) \leq \text{rad}(G) + \delta$. Furthermore, since $2\text{rad}(G) \geq d_G(v, t) \geq 2\text{rad}(G) - 4\delta - 1$ (by (c)), $\text{rad}(G) \geq d_G(t, c) = \lceil \frac{d_G(v, t)}{2} \rceil \geq \text{rad}(G) - 2\delta$. Hence,

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

implying

$$\text{ecc}_G(c) \leq d_G(w, c) + \text{ecc}_G(w) \leq \text{rad}(G) + 3\delta.$$

Let now x be an arbitrary vertex from $C^k(G)$, i.e., $\text{ecc}_G(x) \leq \text{rad}(G) + k$, 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 v and t . Let m_x be a point on $[t, v]$ which is at distance $(x|t)_v = \frac{1}{2}(d_G(x, v) + d_G(v, t) - d_G(x, t))$ from v and hence at distance $(x|v)_t = \frac{1}{2}(d_G(x, t) + d_G(v, t) - d_G(x, v))$ 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., $d_G(v, c^*) \leq d_G(v, m_x) = (x|t)_v$.

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 $rad(G) + k$, we get

$$\begin{aligned} d_G(x, c^*) &\leq (t|v)_x + \delta + (x|t)_v - d_G(v, c^*) = d_G(v, x) - d_G(v, c^*) + \delta \\ &\leq rad(G) + k - \frac{d_G(v, t)}{2} + \delta \leq k + 3\delta + \frac{1}{2}. \end{aligned}$$

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 $ecc_G(c) \leq rad(G) + 3\delta$, we get

$$\begin{aligned} d_G(x, c) &\geq ecc_G(x) - ecc_G(c) = k + rad(G) - ecc_G(c) \\ &\geq (k + rad(G)) - (rad(G) + 3\delta) = k - 3\delta. \end{aligned}$$

□

Recall that $C^k(G) := \{v \in V : ecc_G(v) = rad(G) + k\}$. Next we show that all vertices of $C^k(G)$ ($0 \leq k \leq diam(G) - rad(G)$) are pairwise at distance at most $2k + 2\delta + 1$ in G . Hence, the diameter of the center $C(G)$ of G is at most $2\delta + 1$.

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

Proof: Let x, y be two vertices of $C^k(G)$ such that $d_G(x, y) = diam(C^k(G))$. 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 $[z, x], [y, z]$ are arbitrary geodesics connecting z with x and y . Let m_z be a point on $[x, y]$ which is at distance $(x|z)_y = \frac{1}{2}(d_G(x, y) + d_G(z, y) - d_G(x, z))$ from y and hence at distance $(y|z)_x = \frac{1}{2}(d_G(x, y) + d_G(z, x) - d_G(y, z))$ from x . Without loss of generality, we can assume that m is located on $[x, y]$ between y and m_z .

Since $ecc_G(y) \leq rad(G) + k$, we have

$$d_G(m, z) = ecc_G(m) \geq rad(G) - \frac{1}{2} \geq ecc_G(y) - k - \frac{1}{2} \geq d_G(y, z) - k - \frac{1}{2}.$$

On the other hand, by the triangle inequality, we get

$$\begin{aligned} d_G(m, z) &\leq (x|z)_y - d_G(y, m) + \delta + (x|y)_z = d_G(y, z) - d_G(y, m) + \delta \\ &\leq d_G(y, z) - \frac{d_G(x, y)}{2} + \delta. \end{aligned}$$

Hence, $d_G(x, y) \leq 2k + 2\delta + 1$. □

3.2.1 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 δ -thin triangles can be computed in

$O(\delta|E|)$ total time as follows. By Proposition 3(c), if v is a most distant vertex from an arbitrary vertex u and t is a most distant vertex from v , then $d_G(v, t) \geq \text{diam}(G) - 2\delta$. Hence, using at most $O(\delta)$ *breadth-first-searches*, one can generate a sequence of vertices $v := v_1, t := v_2, v_3, \dots, v_k$ 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 $d_G(v, t) \geq \text{diam}(G) - 2\delta$ can be improved at most 2δ times).

Thus, by Proposition 2 and Proposition 3, we get the following additive approximations for the radius and the diameter of a graph with δ -thin triangles.

Corollary 2 *Let $G = (V, E)$ be a graph with δ -thin triangles.*

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

3.2.2 All eccentricities.

In what follows, we will show that all vertex eccentricities of a graph with δ -thin triangles can be also additively approximated in (almost) linear time. It will be convenient, for the middle point m of an edge e of G , to define a *BFS(m)*-tree of G ; it is nothing else than a *BFS(e)*-tree of G rooted at edge e .

Proposition 5 *Let G be a graph with δ -thin triangles.*

- (a) *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 $\lceil \frac{d_G(v, t)}{2} \rceil$ from t and T is a *BFS(c)*-tree of G , then $\text{ecc}_G(x) \leq \text{ecc}_T(x) \leq \text{ecc}_G(x) + 6\delta + 1$.*
- (b) *If c^* is the middle point 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 , $\text{ecc}_G(x) \leq \text{ecc}_T(x) \leq \text{ecc}_G(x) + 2\delta$.*

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 3(d) that $\text{ecc}_G(c) \leq \text{rad}(G) + 3\delta$ and $d_G(c, x) \leq k + 3\delta + 1$. Since T is a *BFS(c)*-tree, $d_G(x, c) = d_T(x, c)$ and $\text{ecc}_G(c) = \text{ecc}_T(c)$. Consider a vertex y in G such that $d_T(x, y) = \text{ecc}_T(x)$. We have

$$\begin{aligned}
 \text{ecc}_T(x) &= d_T(x, y) \leq d_T(x, c) + d_T(c, y) \\
 &\leq d_G(x, c) + \text{ecc}_T(c) = d_G(x, c) + \text{ecc}_G(c) \\
 &\leq k + 3\delta + 1 + \text{rad}(G) + 3\delta = \text{rad}(G) + k + 6\delta + 1 \\
 &= \text{ecc}_G(x) + 6\delta + 1.
 \end{aligned}$$

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

(b) Consider an arbitrary vertex x of G and a geodesic triangle $\Delta(x, u, v) := [x, u] \cup [u, v] \cup [v, x]$, where $[u, v]$ is a (u, v) -geodesic containing c^* and $[u, x], [v, x]$ are arbitrary geodesics connecting x with u and v . Let m_x be a point on $[u, v]$ which is at distance $(x|u)_v = \frac{1}{2}(d_G(x, v) + d_G(u, v) - d_G(x, u))$ from v and hence at distance $(x|v)_u = \frac{1}{2}(d_G(x, u) + d_G(v, u) - d_G(x, v))$ from u . Without loss of generality, we can assume that c^* is located on $[u, v]$ between v and m_x . We have (see Fig. 3),

$$\begin{aligned} d_G(x, c^*) &\leq (u|v)_x + \delta + d_G(m_x, c^*) \\ &= (u|v)_x + \delta + d_G(u, c^*) - (v|x)_u \\ &= (u|v)_x + \delta + \frac{d_G(v, u)}{2} - (v|x)_u, \end{aligned}$$

and

$$\text{ecc}_G(x) \geq d_G(x, v) = (u|v)_x + (u|x)_v.$$

Furthermore, by Proposition 2(a), $\text{ecc}_G(c^*) \leq \frac{d_G(v, u)}{2} + \delta$. Hence,

$$\begin{aligned} \text{ecc}_T(x) - \text{ecc}_G(x) &\leq d_T(x, c^*) + \text{ecc}_T(c^*) - \text{ecc}_G(x) \\ &= d_G(x, c^*) + \text{ecc}_G(c^*) - \text{ecc}_G(x) \\ &\leq (u|v)_x + \delta + \frac{d_G(v, u)}{2} - (v|x)_u + \\ &\quad \frac{d_G(v, u)}{2} + \delta - (u|v)_x - (u|x)_v \\ &= 2\delta + d_G(v, u) - ((v|x)_u + (u|x)_v) \\ &= 2\delta. \end{aligned}$$

□

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 [52, 79]. Thus, by Proposition 5, we get.

Theorem 1 *Every graph $G = (V, E)$ with δ -thin triangles admits an eccentricity (2δ) -approximating spanning tree constructible in $O(\delta|E|)$ time and an eccentricity $(6\delta + 1)$ -approximating spanning tree constructible in $O(|E|)$ time.*

Theorem 1 generalizes recent results from [52, 79] 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 $rad(T)$ of any tree T can be found in linear time.
- (3) For every vertex $v \in V$, $ecc_T(v) = d_T(v, C(T)) + rad(T)$.

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

Theorem 2 *Let $G = (V, E)$ be a graph with δ -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 $ecc_G(v)$ such that $ecc_G(v) \leq \hat{e}(v) \leq ecc_G(v) + 6\delta + 1$.*
- (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 $ecc_G(v)$ such that $ecc_G(v) \leq \hat{e}(v) \leq ecc_G(v) + 2\delta$.*

4 Fast Additive Approximation of All Distances

Here, we will show that if the δ th power G^δ of a graph G with δ -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^δ 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 [51] 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 -chordal graphs with an additive one-sided error of at most $k - 1$ can be found in $O(|V|^2)$ time; all distances in chordal graphs with an additive one-sided error of at most 1 can be found in $O(|V|^2)$ time and the 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 δ -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 often omit the subscript T . Let $P_T(x, s) := (x_q, x_{q-1}, \dots, x_1, s)$ and $P_T(y, s) := (y_p, y_{p-1}, \dots, y_1, s)$ be the paths of T connecting vertices x and y with the root s . By $sl_T(x, y; \lambda)$ we denote the largest index k such that $d_G(x_k, y_k) \leq \lambda$ (the λ separation level). Our method is based on the following simple fact.

Proposition 6 *For every vertices x and y of a graph G with δ -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 $k = sl_T(x, y; \delta)$.

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 δ -thin, $sl_T(x, y; \delta) \geq (x|y)_s - \frac{1}{2}$. Hence, $h_T(x) - sl_T(x, y; \delta) \leq (s|y)_x + \frac{1}{2}$ and $h_T(y) - sl_T(x, y; \delta) \leq (s|x)_y + \frac{1}{2}$. As $d_G(x, y) = (s|y)_x + (s|x)_y$, we get $d_G(x, y) \geq h_T(x) - sl_T(x, y; \delta) + h_T(y) - sl_T(x, y; \delta) - 1$. \square

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 $\sigma := [v_1, v_2, \dots, v_n = s]$ 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^δ is known in advance (i.e., its adjacency matrix is given) for a graph G with δ -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 σ from 1 to n . For each current vertex x we show that the values $\widehat{d}(x, y) := h_T(x) + h_T(y) - 2sl_T(x, y; \delta) + \delta$ for all vertices y with $y > x$ can be computed in $O(|V|)$ total time. By Proposition 6,

$$d_G(x, y) \leq \widehat{d}(x, y) \leq d_G(x, y) + \delta + 1.$$

The values $\widehat{d}(x, y)$ 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 $L_i = \{v \in V : d_G(v, s) = i\}$. In the procedure, S_u represents vertices of a subtree of T rooted at u .

- (01) set $q := h(x)$
- (02) define a set $S_u := \{u\}$ for each vertex $u \in L_q, u > x$, and denote this family of sets by \mathcal{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^δ)
- (07) **then for** every $v \in S_u$
- (08) set $\widehat{d}(x, v) := h(x) + h(v) - 2k + \delta$ and remove S_u from \mathcal{F}
- (09) **endfor**
- (10) **endfor**
- (11) /* update \mathcal{F} for the next iteration */
- (12) **if** $k > 0$ **then**
- (13) **for** each vertex $u \in L_{k-1}$

- (14) combine all sets $S_{u_1}, \dots, S_{u_\ell}$ from \mathcal{F} ($\ell \geq 0$), such that
 $p_T(u_1) = \dots = p_T(u_\ell) = u$, into one new set
 $S_u := \{u\} \cup S_{u_1} \cup \dots \cup S_{u_\ell}$ /* when $\ell = 0$, $S_u := \{u\}$ */
- (15) **endfor**
- (16) **endfor**
- (17) set also $\widehat{d}(x, s) := h(x)$.

Thus, we have the following result.

Theorem 3 *Let $G = (V, E)$ be a graph with δ -thin triangles. Given G^δ , 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^δ is given in advance, we can use any known fast constant-factor approximation algorithm that in total $T(|V|)$ -time computes for every pair of vertices x, y of G a value $\widetilde{d}(x, y)$ such that $d_G(x, y) \leq \widetilde{d}(x, y) \leq \alpha d_G(x, y) + \beta$. We can show that, using such an algorithm as a preprocessing step, the distances in a graph G with δ -thin triangles can be additively approximated with an additive one-sided error of at most $\alpha\delta + \beta + 1$ 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 $\widetilde{d}(x, y)$ such that

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

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

Proposition 7 *For every vertices x and y of a graph G with δ -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 $k = \widetilde{sl}_T(x, y; 2\rho + 1)$.

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 $[x, s] = P_T(x, s) = (x_q, x_{q-1}, \dots, x_1, s)$ and $[y, s] = P_T(y, s) = (y_p, y_{p-1}, \dots, y_1, s)$, for each $i \leq (x|y)_s$, $d_G(x_i, y_i) \leq \delta \leq \rho$ and, hence, $\widetilde{d}(x_i, y_i) \leq 2\rho + 1$ holds. Therefore, $\widetilde{sl}_T(x, y; 2\rho + 1) \geq (x|y)_s - \frac{1}{2}$. \square

Let ρ be any integer greater than or equal to δ . By replacing in our earlier procedure lines (06) and (08) with

- (06)' **if** $\widetilde{d}(u, x_k) \leq 2\rho + 1$ **then**
- (08)' set $\widehat{d}(x, v) := h(x) + h(v) - 2k + 2\rho + 1$ and remove S_u from \mathcal{F}

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

$$\begin{aligned} d_G(x, y) &\leq h_T(x) + h_T(y) - 2\widetilde{sl}_T(x, y; 2\rho + 1) + d_G(x_k, y_k) \\ &\leq h_T(x) + h_T(y) - 2\widetilde{sl}_T(x, y; 2\rho + 1) + \widetilde{d}(x_k, y_k) \\ &\leq h_T(x) + h_T(y) - 2\widetilde{sl}_T(x, y; 2\rho + 1) + 2\rho + 1 \\ &= \widehat{d}(x, y) \end{aligned}$$

and

$$\begin{aligned} \widehat{d}(x, y) &= h_T(x) + h_T(y) - 2\widetilde{sl}_T(x, y; 2\rho + 1) + 2\rho + 1 \\ &\leq d_G(x, y) + 2\rho + 2. \end{aligned}$$

Thus, we have the following result:

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

- (a) *If the value of δ 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 ρ of δ 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 [31] 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 ρ^* of the thinness of G such that $\delta \leq \rho^* \leq 8\delta + 12k + 4$, proving a $\tilde{O}(|V|^2)$ -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 2,4.

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

Type	Network	Ref.	$ V $	$ E $	$ C(G) $	$rad(G)$	$diam(G)$	$diam_G(C(G))$	connected?	$\delta(G)$
Social	DUTCH-ELITE	[17]	3621	4310	3	12	22	4	n	5
	FACEBOOK	[74]	4039	88234	1	4	8	0	y	1.5
	EVA	[17]	4475	4664	15	10	18	3	y	3.5
	SLASHDOT	[72]	77360	905468	1	6	12	0	y	*1.5
	LOANS	[70]	89171	3394979	29350	5	8	4	y	
	TWITTER	[41]	465017	834797	755	5	8	4	y	
Commun.	EMAIL-VIRGLI	[61]	1133	5451	215	5	8	4	y	2
	EMAIL-ENRON	[72]	33696	180811	248	7	13	2	y	
	EMAIL-EU	[71]	224832	680720	1	7	14	0	y	
	WIKITALK-CHINA	[85]	1217365	3391055	17	4	8	2	y	
Biologic	CS-METABOLIC	[55]	453	4596	17	4	7	2	y	1.5
	SC-PPI	[65]	1458	1948	48	11	19	6	n	3.5
	YEAST-PPI	[28]	2224	6609	57	6	11	4	n	2.5
	HOMO-PI	[83]	16635	115364	135	5	10	2	n	2
Internet	AS-GRAPH-1	[1]	3015	5156	32	5	9	2	y	2
	AS-GRAPH-2	[1]	4885	9276	531	6	11	4	n	3
	AS-GRAPH-3	[1]	5357	10328	10	5	9	2	y	2
	ROUTEVIEW	[7]	10515	21455	2	5	10	2	n	2.5
	AS-CAIDA	[4]	26475	53381	2	9	17	1	y	2.5
	ITDK	[5]	190914	607610	155	14	26	4	y	
P2P	GNUTELLA-06	[71]	8717	31525	338	6	10	5	n	3
	GNUTELLA-24	[71]	26498	65359	1	6	11	0	y	3
	GNUTELLA-30	[71]	36646	88303	602	7	11	6	n	*2.5
	GNUTELLA-31	[71]	62561	295756	55	7	11	5	n	*2.5
Web	WEB-STANFORD	[72]	255265	2234572	1	82	164	0	y	*7
	WEB-NOTREDAM	[13]	325729	1497134	12	23	46	2	n	*2
	WEB-BERKSTAN	[72]	654782	7600595	1	104	208	0	y	*7
Co-p.	AMAZON-1	[93]	334863	925872	21	24	47	3	n	
	AMAZON-2	[93]	400727	3200440	194	11	20	5	n	
Infrastr.	ROAD-EURO	[84]	1039	1305	1	31	62	0	y	7.5
	OPENFLIGHT	[6]	3397	19231	21	7	13	2	y	2
	POWER-GRID	[88]	4941	6594	1	23	46	0	y	10
	ROAD-PA	[72]	1087562	3083028	2	402	794	1	y	*195.5

Table 1: Statistics of the analyzed networks: $|V|$ is the number of vertices, $|E|$ is the number of edges; $|C(G)|$ is the number of central vertices; $rad(G)$ is the graph's radius; $diam(G)$ is the graph's diameter; $diam_G(C(G))$ is the diameter of the graph's center; "connected?" indicates whether or not the center of the graph is connected; $\delta(G)$ is the graph's hyperbolicity. Hyperbolicity values marked with asterisks are approximate.

Large Network Dataset Collection (SNAP) and the Koblenz Network Collection (KONECT), and are available at [2] and [3]. 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 δ of majority of networks². Computing the hyperbolicity of a graph is computationally expensive; therefore, we provide the exact δ values for the smaller networks (those with $|V| \leq 30K$) in our dataset (in some cases, the algorithm proposed in [42] was used). For some larger networks, the approximated δ -hyperbolicity values listed in Table 1 are as reported in [67]³. 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 of 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. Here, by a *BFS* sweep of a graph G starting at a vertex s , we mean a traversal of G which visits all vertices of G in a *breadth-first-search* order starting from s .

In column four of Table 2, we report for each graph G the difference between $2rad(G)$ and $d_G(u, v)$. Proposition 2(c) says that the difference must be at most $2\delta + 1$, where δ is the thinness of geodesic triangles in G . Actually, for large number (27 out of 33) of graphs in our dataset, the difference is at most two. Five other graphs have the difference equal to 3, and only ROAD-PA network has the difference equal to 10. We have $d_G(u, v) = diam(G)$ for 27 graphs in our dataset, including ROAD-PA network whose geodesic triangles thinness is at least 196. For remaining six graphs $d_G(u, v) = diam(G) - 1$ 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

²All δ -hyperbolicity values listed in Table 1 were computed using Gromov's four-point condition definition. As mentioned in [59, 60], geodesic triangles of geodesic δ -hyperbolic spaces are 4δ -thin.

³For 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.

Network	No. of BFS sweeps	$d_G(u, v)$	$2rad(G) - d_G(u, v)$	$ecc_G(c)$	$ecc_G(c) - rad(G)$	$d_G(c, C(G))$	$\min i : B(c, i) \supseteq C(G)$	k_{max}	k_{avg}
DUTCH-ELITE	2	22	2	13	1	1	3	6	2.35
FACEBOOK	2	8	0	4	0	0	0	2	0.686
EVA	2	18	2	10	0	0	2	2	0.571
SLASHDOT	2	11	1	7	1	2	2	3	1.777
LOANS	2	7	3	5	0	0	3	3	2.06
TWITTER	2	8	2	6	1	1	3	4	2.569
EMAIL-VIRGILI	2	7	3	6	1	1	3	4	2.729
EMAIL-ENRON	2	13	1	7	0	0	2	2	0.906
EMAIL-EU	2	14	0	7	0	0	0	2	0.002
WIKITALK-CHINA	2	7	1	5	1	1	2	3	2.076
CE-METABOLIC	2	7	1	5	1	1	2	3	1.982
SC-PPI	3	19	3	12	1	2	6	3	0.981
YEAST-PPI	3	11	1	6	0	0	3	3	1.872
HOMO-PI	2	10	0	5	0	0	2	2	0.747
AS-GRAPH-1	2	8	2	6	1	1	2	3	1.791
AS-GRAPH-2	3	11	1	6	0	0	3	3	1.124
AS-GRAPH-3	2	9	1	5	0	0	2	2	0.828
ROUTEVIEW	2	10	0	5	0	0	2	2	0.329
AS-CAIDA	2	17	1	9	0	0	1	0	0
ITDK	2	26	2	15	1	1	3	4	2.108
GNUTELLA-06	2	10	2	6	0	0	4	4	2.507
GNUTELLA-24	2	10	2	7	1	1	1	5	2.697
GNUTELLA-30	2	11	3	7	0	0	5	5	3.167
GNUTELLA-31	2	11	3	8	1	2	5	6	4.176
WEB-STANFORD	2	164	0	82	0	0	0	28	0.006
WEB-NOTREDAM	2	46	0	23	0	0	2	2	0.935
WEB-BERKSTAN	2	208	0	104	0	0	0	22	0.002
AMAZON-1	2	47	1	24	0	0	2	6	0.991
AMAZON-2	2	20	2	12	1	2	5	6	3.735
ROAD-EURO	2	62	0	31	0	0	0	8	0.135
OPENFLIGHT	2	13	1	8	1	1	2	3	1.879
POWER-GRID	4	46	0	28	5	8	8	13	5.735
ROAD-PA	3	794	10	415	13	44	45	98	23.339

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 (see Proposition 2). "No. of BFS sweeps" 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, $k_{max} := \max_{x \in V} k(x)$ and $k_{avg} := \frac{1}{n} \sum_{x \in V} k(x)$.

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 $\text{ecc}_G(c) - \text{rad}(G) > 2$ (POWER-GRID and ROAD-PA), 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 1, for each graph $G = (V, E)$ in our dataset, we constructed an arbitrary $\text{BFS}(c)$ -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 for it $\text{BFS}(c)$ -tree T_1 , we computed $k_{\max} := \max_{v \in V} \{\text{ecc}_{T_1}(v) - \text{ecc}_G(v)\}$ (*maximum distortion*) and $k_{\text{avg}} := \frac{1}{n} \sum_{v \in V} \text{ecc}_{T_1}(v) - \text{ecc}_G(v)$ (*average distortion*). For most graphs (see Table 2), the value of k_{\max} is small: $k_{\max} = 0$ for one graph, $k_{\max} = 2$ for eight graphs, $k_{\max} = 3$ for nine graphs, $k_{\max} = 4$ for four graphs, $k_{\max} = 5$ for two graphs, and $k_{\max} > 5$ for nine graphs. Also, the average distortion k_{avg} is much smaller than k_{\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_{\max} , close inspection reveals that only small percent of vertices achieve this maximum. For example, in graph WEB-STANFORD, $k_{\max} = 28$ was only achieved by 17 vertices. The distributions of the values of $k(v) := \text{ecc}_{T_1}(v) - \text{ecc}_G(v)$ of all graphs are listed in Table 8 (see Appendix).

Similar experiments were performed following Proposition 3. For each graph G in our dataset, we picked a random vertex $u \in V$ and a random vertex $v \in F(u)$. Then, we identified in a randomly picked (u, v) -geodesic a vertex w at distance $\text{rad}(G)$ 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 3 that $\text{ecc}_G(v) \geq \text{diam}(G) - 2\delta \geq 2\text{rad}(G) - 4\delta - 1$ and $\text{ecc}_G(w) \leq \text{rad}(G) + \delta$. Our experimental results are better than these theoretical bounds. In Table 3, we list eccentricities of v and w for each graph. In almost all graphs, the eccentricity of v is equal to the diameter $\text{diam}(G)$. Only four graphs have $\text{ecc}_G(v) = \text{diam}(G) - 1$ and one graph (ROAD-PA) has $\text{ecc}_G(v) > \text{diam}(G) - 1$. Vertex w is central for 21 graphs, has eccentricity equal to $\text{rad}(G) + 1$ for 10 graphs, has eccentricity equal to $\text{rad}(G) + 2$ for one graph, and only for one remaining graph (ROAD-PA network, which has large hyperbolicity) its eccentricity is equal to $\text{rad}(G) + 15$. 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 $2\text{rad}(G) - \text{ecc}_G(w) = 32$ and $d_G(w, C(G)) = 21$.

For every graph $G = (V, E)$ in our dataset, we constructed also an arbitrary $\text{BFS}(w)$ -tree $T_2 = (V, E')$, rooted at vertex w , and analyzed how well T_2 pre-

Network	v			$C(G)$		k_{max}	k_{avg}
	$ecc_G(v)$	$2rad(G) - ecc_G(v)$	$ecc_G(w)$	$d_G(w, C(G))$	$\min i : B(w, i) \supseteq C(G)$		
DUTCH-ELITE	22	2	12	0	4	6	2.431
FACEBOOK	8	0	5	3	3	3	0.704
EVA	18	2	11	1	3	2	0.572
SLASHDOT	11	1	7	2	2	3	1.88
LOANS	7	3	5	0	3	3	2.031
TWITTER	8	2	5	0	3	3	1.821
EMAIL-VIRGILI	7	3	5	0	4	4	1.932
EMAIL-ENRON	13	1	7	0	2	2	0.903
EMAIL-EU	14	0	7	0	0	2	0.002
WIKITALK-CHINA	8	0	5	1	2	3	1.791
CE-METABOLIC	7	1	4	0	1	1	0.349
SC-PPI	19	3	12	1	6	7	4.196
YEAST-PPI	11	1	7	1	3	4	2.558
HOMO-PI	9	1	5	0	2	2	0.612
AS-GRAPH-1	9	1	5	0	2	2	0.887
AS-GRAPH-2	11	1	6	0	3	2	0.833
AS-GRAPH-3	9	1	5	0	2	2	0.312
ROUTEVIEW	10	0	5	0	2	2	0.329
AS-CAIDA	17	1	9	0	1	0	0
ITDK	26	2	15	1	3	5	2.702
GNUTELLA-06	10	2	7	1	5	5	3.543
GNUTELLA-24	11	1	8	3	3	6	4.475
GNUTELLA-30	11	3	8	1	5	6	4.034
GNUTELLA-31	11	3	8	1	5	6	4.251
WEB-STANFORD	164	0	82	0	0	28	0.006
WEB-NOTREDAM	46	0	23	0	2	2	0.935
WEB-BERKSTAN	208	0	104	0	0	22	0.002
AMAZON-1	47	1	24	0	3	7	0.919
AMAZON-2	20	2	11	0	5	5	2.03
ROAD-EURO	62	0	31	0	0	8	0.135
OPENFLIGHT	13	1	7	0	2	2	0.641
POWER-GRID	46	0	23	0	0	4	1.409
ROAD-PA	772	32	417	21	22	80	22.545

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 (see Proposition 3), 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 $k_{avg} := \frac{1}{n} \sum_{x \in V} k(x)$.

serves 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 9 (see Appendix).

Network	$diam(G)$	$diam(T_1)$			$diam(T_2)$			$diam(T_3)$		
		$k_{max}^{T_1}$	$k_{avg}^{T_1}$		$k_{max}^{T_2}$	$k_{avg}^{T_2}$		$k_{max}^{T_3}$	$k_{avg}^{T_3}$	
DUTCH-ELITE	22	24	6	2.35	24	6	2.431	24	6	2.083
FACEBOOK	8	8	2	0.686	9	3	0.704	8	2	0.686
EVA	18	19	2	0.571	19	2	0.572	19	2	0.571
SLASHDOT	12	14	3	1.777	14	3	1.88	12	2	0.701
LOANS	8	10	3	2.06	10	3	2.031	10	3	2.081
TWITTER	8	11	4	2.569	10	3	1.821	10	4	1.856
EMAIL-VIRGILI	8	11	4	2.729	10	4	1.932	10	4	1.906
EMAIL-ENRON	13	13	2	0.906	14	2	0.903	14	2	1.735
EMAIL-EU	14	14	2	0.002	14	2	0.002	14	2	0.002
WIKITALK-CHINA	8	9	3	2.076	9	3	1.791	8	2	0.777
CE-METABOLIC	7	9	3	1.982	8	1	0.349	8	2	1.185
SC-PPI	19	20	3	0.981	23	7	4.196	22	6	3.163
YEAST-PPI	11	12	3	1.872	13	4	2.558	12	3	1.872
HOMO-PI	10	10	2	0.747	10	2	0.612	10	2	0.747
AS-GRAPH-1	9	11	3	1.791	10	2	0.887	10	2	0.886
AS-GRAPH-2	11	11	3	1.124	11	2	0.833	12	3	1.272
AS-GRAPH-3	9	10	2	0.828	10	2	0.312	10	2	0.312
ROUTEVIEW	10	10	2	0.329	10	2	0.329	10	2	0.329
AS-CAIDA	17	17	0	0	17	0	0	17	0	0
ITDK	26	29	4	2.108	29	5	2.702	28	3	1.385
GNUTELLA-06	10	12	4	2.507	13	5	3.543	12	4	2.507
GNUTELLA-24	11	14	5	2.697	16	6	4.475	12	3	0.863
GNUTELLA-30	11	14	5	3.167	16	6	4.034	14	5	3.295
GNUTELLA-31	11	16	6	4.176	16	6	4.251	14	5	2.669
WEB-STANFORD	164	164	28	0.006	164	28	0.006	164	28	0.006
WEB-NOTREDAM	46	46	2	0.935	46	2	0.935	46	2	0.017
WEB-BERKSTAN	208	208	22	0.002	208	22	0.002	208	22	0.002
AMAZON-1	47	47	6	0.991	48	7	0.919	47	7	1.205
AMAZON-2	20	23	6	3.735	22	5	2.03	22	4	1.274
ROAD-EURO	62	62	8	0.135	62	8	0.135	62	8	0.135
OPENFLIGHT	13	15	3	1.879	14	2	0.641	14	2	0.704
POWER-GRID	46	51	13	5.735	46	4	1.409	46	4	1.409
ROAD-PA	794	814	98	23.339	830	80	22.545	803	46	10.64

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 $c^* \in C(G)$.

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).

We also compared the performance of our base algorithm with a popular heuristic k -SumSweep (see [22, 23]) for estimating radii and diameters of large-scale graphs. In our algorithm, for a given graph G , we find with a few *BFS* sweeps a mutually distant pair of vertices u and v and a middle vertex c of an arbitrary shortest path connecting u and v , and report $d_G(u, v)$ as an estimate for the diameter of G and $ecc_G(c)$ as an estimate for the radius of G . k -SumSweep heuristic starts with an arbitrary vertex s and sets $F := \{s\}$. It runs for k iterations ($k \in \mathbb{N}$ is a parameter of the heuristic). At each iteration a new vertex s that maximizes $\sum_{v \in F} d_G(s, v)$ is identified and put into F . After k iterations, $\max\{ecc_G(s) : s \in F\}$ is reported as an estimate for the diameter of G and $ecc_G(c)$, where c is a vertex minimizing $\sum_{v \in F} d_G(c, v)$, is reported as an estimate for the radius of G .

In Table 5, we show our experimental results for large graphs (up-to 1.2M vertices). For these graphs, the experiments were performed on a PC with Intel Core i7 processor @ 3GHz and 8GB memory. Since our algorithm needed from two to four *BFS* sweeps to find a mutually distant pair of vertices (see Table 2), we run k -SumSweep heuristic for three iterations only (to properly compare the running times and the quality of outputs). We can see that already three iterations of k -SumSweep heuristic require slightly more time to terminate than our algorithm requires. Both algorithms succeed in finding the exact diameter of most of those graphs (except that for EMAIL-VIRGILI the output of our algorithm is one unit short from the diameter and for ROAD-PA the output of 3-SumSweep heuristic is a few units short from the diameter). However, in given amounts of time, our algorithm often finds better estimates for radii than 3-SumSweep heuristic does. We get same estimates for 12 graphs, our algorithm gives better estimates for 14 graphs, and 3-SumSweep heuristic gives slightly better estimates for 7 graphs. The exact radius is found by our algorithm in 22 graphs and by 3-SumSweep heuristic in 16 graphs.

To see the performance of both methods on very large graphs, we run them on two networks obtained from the Koblenz Network Collection (KONECT) [3]. The first network LIVEJOURNAL has 5.2M nodes and 49M edges and its diameter is 23. The second network ORKUT has 3M nodes and 117M edges and its diameter is 10. For these very large graphs, the experiments were performed on a cluster

Network	$rad(G)$ $diam(G)$		our Alg			3-SumSweep		
	rad'	$diam'$	time	rad'	$diam'$	time	rad'	$diam'$
DUTCH-ELITE	12	22	13	22	0.003	13	22	0.032
FACEBOOK	4	8	4	8	0.049	5	8	0.087
EVA	10	18	10	18	0.016	10	18	0.033
SLASHDOT	6	12	7	12	0.259	7	12	0.286
LOANS	5	8	5	8	0.745	5	8	0.827
TWITTER	5	8	6	8	0.569	5	8	0.659
EMAIL-VIRGILI	5	8	5	7	0.004	5	8	0.022
EMAIL-ENRON	7	13	7	13	0.081	8	13	0.169
EMAIL-EU	7	14	7	14	0.372	8	14	0.522
WIKI-CHINA	4	8	4	8	1.723	5	8	2.19
CE-METABOLIC	4	7	4	7	0.005	4	7	0.02
PPI	11	19	12	19	0.011	11	19	0.017
YEAST-PPI	6	11	6	11	0.004	7	11	0.028
HOMO-PI	5	10	5	10	0.054	6	10	0.139
AS-GRAPH-1	5	9	5	9	0.021	5	9	0.055
AS-GRAPH-2	6	11	6	11	0.043	6	11	0.048
AS-GRAPH-3	5	9	6	9	0.012	5	9	0.058
ROUTEVIEW	5	10	5	10	0.061	5	10	0.067
AS-CAIDA	9	17	9	17	0.046	12	17	0.239
ITDK	14	26	15	26	0.345	14	26	0.401
GNUTELLA-06	6	10	6	10	0.037	6	10	0.075
GNUTELLA-24	6	11	8	11	0.082	6	11	0.169
GNUTELLA-30	7	11	8	11	0.068	7	11	0.17
GNUTELLA-31	7	11	8	11	0.145	8	11	0.214
WEB-STANFORD	82	164	82	164	0.763	98	164	0.991
WEB-NOTREDAM	23	46	23	46	0.324	24	46	0.476
WEB-BERKSTAN	104	208	104	208	1.735	122	208	1.918
AMAZON-1	24	47	24	47	0.534	26	47	0.668
AMAZON-2	11	20	11	20	1.156	11	20	1.352
ROAD-EURO	31	62	31	62	0.009	36	62	0.017
OPENFLIGHT	7	13	8	13	0.032	7	13	0.041
POWER-GRID	23	46	23	46	0.031	27	46	0.07
ROAD-PA	402	794	415	794	1.065	469	785	1.522

Table 5: Performance of our algorithm and k -SumSweep heuristic on large graphs. The k -SumSweep heuristic is run for $k = 3$. Values rad' and $diam'$ are the estimates for the radius and the diameter output by corresponding methods. Running times are given in seconds.

computer with larger main memory; we used Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz with 20480 KB cache and 94GB main memory. For both graphs, our algorithm found a mutually distant pair of vertices using only three *BFS* sweeps. So, we run *k*-SumSweep heuristic on them first for $k = 3$ and then for $k = 4$ for comparison. Our algorithm was again faster (see Table 6). It succeeded to find the exact diameter and the exact radius for LIVEJOURNAL in about 5.23 minutes. Both, 3-SumSweep and 4-SumSweep, heuristics computed the exact diameter for LIVEJOURNAL but were one unit off from the exact radius and required more time. For ORKUT, 3-SumSweep and 4-SumSweep, both were one unit off from the exact radius and one unit off from the exact diameter. Our algorithm was faster and found the exact radius of ORKUT but was also one unit off from the exact diameter. Note that the exact radii for those two graphs were previously unknown to us but since for any graph G , $diam(G) \leq 2rad(G)$, the radii of those graphs cannot be smaller than 12 and 5, respectively.

Network	$diam(G)$	our Alg			3-SumSweep			4-SumSweep		
		rad'	$diam'$	time	rad'	$diam'$	time	rad'	$diam'$	time
LIVEJOURNAL	23	12	23	313.77	13	23	338.11	13	23	418.82
ORKUT	10	5	9	1186.23	6	9	1311.91	6	9	1545.74

Table 6: Performance of our algorithm and *k*-SumSweep heuristic on two very large graphs. The *k*-SumSweep heuristic is run for $k = 3$ and $k = 4$. Values rad' and $diam'$ are the estimates for the radius and the diameter output by corresponding methods. Running times are given in seconds.

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 $\delta := 0, 1, 2, \dots$, we computed an estimate $\widehat{d}_\delta(x, y)$ on $d_G(x, y)$ and the error $\Delta_{xy}(\delta) = \widehat{d}_\delta(x, y) - d_G(x, y)$ for all $x, y \in V$. In Table 7, we report $\Delta_{max}(\delta) = \max_{x, y \in V} \Delta_{xy}(\delta)$ and $\Delta_{avg}(\delta) = \frac{1}{n^2} \sum_{x, y \in V} \Delta_{xy}(\delta)$ for the smallest δ 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

Network	$diam(G)$	$rad(G)$	δ	$\Delta_{max}(\delta)$	$\Delta_{avg}(\delta)$	$ecc(s)$
DUTCH-ELITE	22	12	8	8	0.177	16
FACEBOOK	8	4	2	2	0.169	6
EVA	18	10	6	6	0.044	12
SLASHDOT*	12	6	4	2	0.028	8
LOANS*	8	5	3	3	0.213	6
TWITTER*	8	5	3	3	0.156	6
EMAIL-VIRGILI	8	5	3	4	0.39	6
EMAIL-ENRON	13	7	4	4	0.06	9
EMAIL-EU*	14	7	3	2	0.005	10
CE-METABOLIC	7	4	2	3	0.125	4
SC-PPI	19	11	6	6	0.19	13
YEAST-PPI	11	6	4	4	0.239	8
HOMO-PI	10	5	3	3	0.02	7
AS-GRAPH-1	9	5	3	4	0.061	8
AS-GRAPH-2	11	6	4	4	0.034	8
AS-GRAPH-3	9	5	4	3	0.035	9
ROUTEVIEW	10	5	4	4	0.038	6
AS-CAIDA	17	9	3	4	0.022	14
ITDK*	26	14	5	4	0.15	19
GNUTELLA-06	9	6	5	4	0.331	8
GNUTELLA-24	11	6	6	6	0.128	9
GNUTELLA-30*	11	7	6	5	0.439	8
GNUTELLA-31*	11	7	6	5	0.386	9
ROAD-EURO	62	31	21	11	0.927	39
OPENFLIGHT	13	7	3	4	0.029	10
POWER-GRID	46	23	17	17	0.518	38

Table 7: Distance approximations: for every $x, y \in V$, $\Delta_{xy}(\delta) = \widehat{d}_\delta(x, y) - d_G(x, y)$; $\Delta_{max}(\delta) = \max_{x,y \in V} \Delta_{xy}(\delta)$; $\Delta_{avg}(\delta) = \frac{1}{n^2} \sum_{x,y \in V} \Delta_{xy}(\delta)$; δ is defined as the smallest δ ($0 \leq \delta \leq diam(G)$) such that $\Delta_{max}(\delta) \leq \delta + 1$. Due to large sizes of some networks, the values of $\Delta_{max}(\delta)$ and $\Delta_{avg}(\delta)$ 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.

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

We would like to thank anonymous reviewers of the journal and short versions of the paper for many useful suggestions and comments. We are very grateful to Abdulhakeem Mohammed for performing experiments for Table 6 on Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz cluster computer.

References

- [1] <http://web.archive.org/web/20060506132945/>.
- [2] <https://snap.stanford.edu/data/>.
- [3] <http://konect.uni-koblenz.de/networks/>.
- [4] Center for applied internet data analysis. <http://www.caida.org/data/as-relationships>.
- [5] Center for applied internet data analysis. <http://www.caida.org/data/internet-topology-data-kit>.
- [6] Openflights network dataset – konect. October 2016.
- [7] University of Oregon route-views project. <http://www.routeviews.org/>.
- [8] A. Abboud, F. Grandoni, and V. V. Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *Proceedings of the twenty-sixth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1681–1697. SIAM, 2014. doi:10.1137/1.9781611973730.112.
- [9] A. Abboud, V. V. Williams, and J. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proceedings of the twenty-seventh annual ACM-SIAM Symposium on Discrete Algorithms*, pages 377–391. SIAM, 2016. doi:10.1137/1.9781611974331.ch28.
- [10] M. Abu-Ata and F. F. Dragan. Metric tree-like structures in real-world networks: an empirical study. *Networks*, 67(1):49–68, 2016. doi:10.1002/net.21631.
- [11] A. B. Adcock, B. D. Sullivan, and M. W. Mahoney. Tree-like structure in large social and information networks. In *2013 IEEE 13th International Conference on Data Mining*, pages 1–10. IEEE, 2013. doi:10.1109/icdm.2013.77.
- [12] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM Journal on Computing*, 28(4):1167–1181, 1999. doi:10.1137/s0097539796303421.
- [13] R. Albert, H. Jeong, and A.-L. Barabási. Internet: Diameter of the world-wide web. *Nature*, 401(6749):130, 1999. doi:10.1038/43601.
- [14] J. Alonso, T. Brady, D. Cooper, V. Ferlini, M. Lustig, M. Mihalik, M. Shapiro, and H. Short. Notes on word hyperbolic groups, group theory from a geometrical viewpoint, ICTP Trieste 1990 (E. Ghys, A. Haefliger, A. Verjovsky, eds.), 1991. doi:10.1142/9789814539746.

- [15] H. Alrasheed. Structural properties in δ -hyperbolic networks: Algorithmic analysis and implications. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 299–303. International World Wide Web Conferences Steering Committee, 2016. doi:10.1145/2872518.2888602.
- [16] H. Alrasheed and F. F. Dragan. Core-periphery models for graphs based on their δ -hyperbolicity: An example using biological networks. *Journal of Algorithms & Computational Technology*, 11(1):40–57, 2017. doi:10.1007/978-3-319-16112-9_7.
- [17] V. Batagelj and A. Mrvar. Pajek datasets. (2006). <http://vlado.fmf.uni-lj.si/pub/networks/data/>.
- [18] B. Ben-Moshe, B. Bhattacharya, Q. Shi, and A. Tamir. Efficient algorithms for center problems in cactus networks. *Theoretical Computer Science*, 378(3):237–252, 2007. doi:10.1016/j.tcs.2007.02.033.
- [19] P. Berman and S. P. Kasiviswanathan. Faster approximation of distances in graphs. In *Workshop on Algorithms and Data Structures*, pages 541–552. Springer, 2007. doi:10.1007/978-3-540-73951-7_47.
- [20] M. Borassi, D. Coudert, P. Crescenzi, and A. Marino. On computing the hyperbolicity of real-world graphs. In *Algorithms-ESA 2015*, pages 215–226. Springer, 2015. doi:10.1007/978-3-662-48350-3_19.
- [21] M. Borassi, P. Crescenzi, and M. Habib. Into the square: On the complexity of some quadratic-time solvable problems. *Electronic Notes in Theoretical Computer Science*, 322:51–67, 2016. doi:10.1016/j.entcs.2016.03.005.
- [22] M. Borassi, P. Crescenzi, M. Habib, W. A. Kosters, A. Marino, and F. W. Takes. Fast diameter and radius BFS-based computation in (weakly connected) real-world graphs: With an application to the six degrees of separation games. *Theoretical Computer Science*, 586:59–80, 2015. doi:10.1016/j.entcs.2016.03.005.
- [23] M. Borassi, P. Crescenzi, and L. Trevisan. An axiomatic and an average-case analysis of algorithms and heuristics for metric properties of graphs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 920–939. SIAM, 2017. doi:10.1137/1.9781611974782.58.
- [24] A. Brandstädt, V. Chepoi, and F. Dragan. Distance approximating trees for chordal and dually chordal graphs. *Journal of Algorithms*, 30(1):166–184, 1999. doi:10.1007/3-540-63397-9_7.
- [25] A. Brandstädt, V. D. Chepoi, and F. F. Dragan. The algorithmic use of hypertree structure and maximum neighbourhood orderings. *Discrete Applied Mathematics*, 82(1-3):43–77, 1998. doi:10.1016/s0166-218x(97)00125-x.

- [26] A. Brandstädt, F. F. Dragan, and F. Nicolai. LexBFS-orderings and powers of chordal graphs. *Discrete Mathematics*, 171(1-3):27–42, 1997. doi:10.1016/s0012-365x(96)00070-2.
- [27] M. R. Bridson and A. Haefliger. *Metric spaces of non-positive curvature Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*, volume 319. Springer-Verlag, Berlin, 1999.
- [28] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, et al. Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic Acids Research*, 31(9):2443–2450, 2003. [Available at: <http://vlado.fmf.uni-lj.si/pub/networks/data/bio/Yeast/Yeast.htm>]. doi:10.1093/nar/gkg340.
- [29] S. Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2143–2152. SIAM, 2017. doi:10.1145/3218821.
- [30] M. Cairo, R. Grossi, and R. Rizzi. New bounds for approximating extremal distances in undirected graphs. In *Proceedings of the twenty-seventh annual ACM-SIAM Symposium on Discrete Algorithms*, pages 363–376. Society for Industrial and Applied Mathematics, 2016. doi:10.1137/1.9781611974331.ch27.
- [31] J. Chalopin, V. Chepoi, F. Dragan, G. Ducoffe, A. Mohammed, and Y. Vaxès. Fast approximation and exact computation of negative curvature parameters of graphs. In *34th International Symposium on Computational Geometry (SoCG 2018)*, page 22, 2018. doi:10.4230/LIPIcs.SocG.2018.22.
- [32] S. Chechik, D. H. Larkin, L. Roditty, G. Schoenebeck, R. E. Tarjan, and V. V. Williams. Better approximation algorithms for the graph diameter. In *Proceedings of the twenty-fifth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1041–1052. Society for Industrial and Applied Mathematics, 2014. doi:10.1137/1.9781611973402.78.
- [33] V. Chepoi and F. Dragan. A linear-time algorithm for finding a central vertex of a chordal graph. In *European Symposium on Algorithms*, pages 159–170. Springer, 1994. doi:10.1007/bfb0049406.
- [34] V. Chepoi and F. Dragan. Finding a central vertex in an HHD-free graph. *Discrete Applied Mathematics*, 131(1):93–111, 2003. doi:10.1016/s0166-218x(02)00419-5.
- [35] V. Chepoi, F. Dragan, B. Estellon, M. Habib, and Y. Vaxès. Diameters, centers, and approximating trees of delta-hyperbolic geodesic spaces and

- graphs. In *Proceedings of the twenty-fourth annual Symposium on Computational Geometry*, pages 59–68. ACM, 2008. doi:10.1145/1377676.1377687.
- [36] V. Chepoi, F. F. Dragan, B. Estellon, M. Habib, Y. Vaxès, and Y. Xi-ang. Additive spanners and distance and routing labeling schemes for hyperbolic graphs. *Algorithmica*, 62(3-4):713–732, 2012. doi:10.1007/s00453-010-9478-x.
- [37] V. Chepoi, F. F. Dragan, M. Habib, Y. Vaxès, and H. Alrasheed. Fast approximation of centrality and distances in hyperbolic graphs. In *Combinatorial Optimization and Applications - 12th International Conference, COCOA 2018, Atlanta, GA, USA, December 15-17, 2018, Proceedings*, pages 3–18, 2018. doi:10.1007/978-3-030-04651-4_1.
- [38] V. Chepoi, F. F. Dragan, and Y. Vaxès. Center and diameter problems in plane triangulations and quadrangulations. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA.*, pages 346–355, 2002. URL: <http://dl.acm.org/citation.cfm?id=545381.545427>.
- [39] V. Chepoi, F. F. Dragan, and Y. Vaxes. Core congestion is inherent in hyperbolic networks. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2264–2279. SIAM, 2017. doi:10.1137/1.9781611974782.149.
- [40] V. Chepoi and B. Estellon. Packing and covering δ -hyperbolic spaces by balls. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 59–73. Springer, 2007. doi:10.1007/978-3-540-74208-1_5.
- [41] M. D. Choudhury, Y. Lin, H. Sundaram, K. S. Candan, L. Xie, and A. Keliher. How does the data sampling strategy impact the discovery of information diffusion in social media? In *Proceedings of the Fourth International Conference on Weblogs and Social Media, ICWSM 2010, Washington, DC, USA, May 23-26, 2010*, 2010. URL: <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/view/1521>.
- [42] N. Cohen, D. Coudert, and A. Lancin. Exact and approximate algorithms for computing the hyperbolicity of large-scale graphs. (RR-8074), Sept. 2012. URL: <https://hal.inria.fr/hal-00735481>.
- [43] D. G. Corneil, F. F. Dragan, M. Habib, and C. Paul. Diameter determination on restricted graph families. *Discrete Applied Mathematics*, 113(2-3):143–166, 2001. doi:10.1007/10692760_16.
- [44] D. G. Corneil, F. F. Dragan, and E. Köhler. On the power of BFS to determine a graph’s diameter. *Networks: An International Journal*, 42(4):209–222, 2003. doi:10.1002/net.10098.

- [45] B. DasGupta, M. Karpinski, N. Mobasher, and F. Yahyanejad. Node expansions and cuts in gromov-hyperbolic graphs. *CoRR*, vol. *abs/1510.08779*, 2015.
- [46] D. Dor, S. Halperin, and U. Zwick. All-pairs almost shortest paths. *SIAM Journal on Computing*, 29(5):1740–1759, 2000. doi:10.1109/sfcs.1996.548504.
- [47] Y. Dourisboure and C. Gavaille. Tree-decompositions with bags of small diameter. *Discrete Mathematics*, 307(16):208–229, 2007. doi:10.1016/j.disc.2005.12.060.
- [48] F. Dragan, M. Habib, and L. Viennot. Revisiting radius, diameter, and all eccentricity computation in graphs through certificates. *CoRR abs/1803.04660*, 2018.
- [49] F. F. Dragan. Centers of graphs and the Helly property (in Russian). *Ph. D. Thesis, Moldova State University*, 1989.
- [50] F. F. Dragan. Almost diameter of a House-Hole-free graph in linear time via LexBFS. *Discrete Applied Mathematics*, 95(1-3):223–239, 1999. doi:10.1016/s0166-218x(99)00077-3.
- [51] F. F. Dragan. Estimating all pairs shortest paths in restricted graph families: a unified approach. *Journal of Algorithms*, 57(1):1–21, 2005. doi:10.1007/3-540-45477-2_11.
- [52] F. F. Dragan, E. Köhler, and H. Alrasheed. Eccentricity approximating trees. *Discrete Applied Mathematics*, 232:142–156, 2017. doi:10.1016/j.dam.2017.07.017.
- [53] F. F. Dragan and F. Nicolai. LexBFS-orderings of distance-hereditary graphs with application to the diametral pair problem. *Discrete Applied Mathematics*, 98(3):191–207, 2000. doi:10.1016/s0166-218x(99)00157-2.
- [54] F. F. Dragan, F. Nicolai, and A. Brandstädt. LexBFS-orderings and powers of graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 166–180. Springer, 1996. doi:10.1016/s0012-365x(96)00070-2.
- [55] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104, 2005. [Available at: <http://deim.urv.cat/~aarenas/data/welcome.htm>]. doi:10.1103/physreve.72.027104.
- [56] D. Dvir and G. Y. Handler. The absolute center of a network. *Networks: An International Journal*, 43(2):109–118, 2004. doi:10.1002/net.10108.

- [57] K. Edwards, S. Kennedy, and I. Saniee. Fast approximation algorithms for p -centers in large δ -hyperbolic graphs. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 60–73. Springer, 2016. doi:10.1007/978-3-319-49787-7_6.
- [58] M. Elkin. Computing almost shortest paths. *ACM Transactions on Algorithms (TALG)*, 1(2):283–323, 2005. doi:10.1145/383962.383983.
- [59] E. Ghys and P. de la Harpe. Les Groupes Hyperboliques d’après M. Gromov. *Progress in Mathematics*, 83, 1990. doi:10.1007/978-1-4684-9167-8_4.
- [60] M. Gromov and S. Gersten. Essays in group theory. *Math. Sci. Res. Inst. Publ.*, 8(75-263):7, 1987.
- [61] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Physical Review E*, 68(6):065103, 2003. doi:10.1103/physreve.68.065103.
- [62] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964. doi:10.1287/opre.12.3.450.
- [63] R. Impagliazzo and R. Paturi. On the complexity of k -sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1109/csc.1999.766282.
- [64] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1109/sfcs.1998.743516.
- [65] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41, 2001. doi:10.1038/35075138.
- [66] C. Jordan. Sur les assemblages de lignes. *J. Reine Angew. Math*, 70(185):81, 1869. doi:10.1515/crll.1869.70.185.
- [67] W. S. Kennedy, I. Saniee, and O. Narayan. On the hyperbolicity of large-scale networks and its estimation. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3344–3351. IEEE, 2016. doi:10.1109/bigdata.2016.7840994.
- [68] D. Koschützki, K. A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski. Centrality indices. In *Network Analysis*, pages 16–61. Springer, 2005. doi:10.1007/978-3-7091-0741-6_9.
- [69] D. Kratsch, H.-O. Le, H. Müller, E. Prisner, and D. Wagner. Additive tree spanners. *SIAM Journal on Discrete Mathematics*, 17(2):332–340, 2003. doi:10.1137/s0895480195295471.

- [70] J. Kunegis. Prosper loans, KONECT, the Koblenz Network Collection, 2016.
- [71] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), Mar. 2007. URL: <http://doi.acm.org/10.1145/1217299.1217301>, doi:10.1145/1217299.1217301.
- [72] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009. doi:10.1080/15427951.2009.10129177.
- [73] M. Madanlal, G. Venkatesan, and C. P. Rangan. Tree 3-spanners on interval, permutation and regular bipartite graphs. *Information Processing Letters*, 59(2):97–102, 1996. doi:10.1016/0020-0190(96)00078-6.
- [74] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12*, pages 539–547, USA, 2012. Curran Associates Inc. URL: <http://dl.acm.org/citation.cfm?id=2999134.2999195>.
- [75] A. Mohammed. *Private communication*, 2017.
- [76] O. Narayan and I. Saniee. Large-scale curvature of networks. *Physical Review E*, 84(6):066108, 2011. doi:10.1103/physreve.84.066108.
- [77] S. Olariu. A simple linear-time algorithm for computing the center of an interval graph. *International Journal of Computer Mathematics*, 34(3-4):121–128, 1990. doi:10.1080/00207169008803870.
- [78] E. Prisner. Distance approximating spanning trees. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 499–510. Springer, 1997. doi:10.1007/bfb0023484.
- [79] E. Prisner. Eccentricity-approximating trees in chordal graphs. *Discrete Mathematics*, 220(1-3):263–269, 2000. doi:10.1016/s0012-365x(00)00030-3.
- [80] L. Roditty and V. Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings of the forty-fifth annual ACM Symposium on Theory of Computing*, pages 515–524. ACM, 2013. doi:10.1145/2488608.2488673.
- [81] L. Roditty and U. Zwick. On dynamic shortest paths problems. *Algorithmica*, 61(2):389–401, 2011. doi:10.1007/s00453-010-9401-5.
- [82] Y. Shavitt and T. Tankel. Hyperbolic embedding of internet graph for distance estimation and overlay construction. *IEEE/ACM Transactions on Networking*, 16(1):25–36, 2008. doi:10.1109/tnet.2007.899021.

- [83] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34(suppl_1):D535–D539, 2006. [Available at: <http://thebiogrid.org/>, release 3.2.99.]. doi:10.1093/nar/gkj109.
- [84] L. Šubelj and M. Bajec. Robust network community detection using balanced propagation. *The European Physical Journal B*, 81(3):353–362, 2011. doi:10.1140/epjb/e2011-10979-2.
- [85] J. Sun, J. Kunegis, and S. Staab. Predicting user roles in social networks using transfer learning with feature transformation. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 128–135. IEEE, 2016. doi:10.1109/icdmw.2016.0026.
- [86] M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM (JACM)*, 51(6):993–1024, 2004. doi:10.1109/sfcs.2001.959898.
- [87] K. Verbeek and S. Suri. Metric embedding, hyperbolic space, and social networks. In *Proceedings of the thirtieth annual Symposium on Computational Geometry*, pages 501–510. ACM, 2014. doi:10.1145/2582112.2582139.
- [88] D. J. Watts and S. H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393(6684):440, 1998. doi:10.1515/9781400841356.
- [89] O. Weimann and R. Yuster. Approximating the diameter of planar graphs in near linear time. *ACM Transactions on Algorithms (TALG)*, 12(1):12, 2016. doi:10.1145/2764910.
- [90] R. Williams. A new algorithm for optimal constraint satisfaction and its implications. In *International Colloquium on Automata, Languages, and Programming*, pages 1227–1237. Springer, 2004. doi:10.1016/j.tcs.2005.09.023.
- [91] V. V. Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*, pages 17–29, 2015. URL: <https://doi.org/10.4230/LIPIcs.IPEC.2015.17>, doi:10.4230/LIPIcs.IPEC.2015.17.
- [92] V. V. Williams and R. Williams. Subcubic equivalences between path, matrix and triangle problems. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 645–654. IEEE, 2010. doi:10.1109/focs.2010.67.
- [93] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015. doi:10.1109/icdm.2012.138.

Appendix

Network	k_{max}	k_{avg}	% of vertices with $k(x) = 0$	% of vertices with $k(x) = 1$	% of vertices with $k(x) = 2$	% of vertices with $k(x) = 3$	% of vertices with $k(x) = 4$	% of vertices with $k(x) = 5$	% of vertices with $k(x) \geq 6$
DUTCH-ELITE	6	2.35	14.9	0	54.3	0	29.1	0	1.7
FACEBOOK	2	0.686	51.9	27.6	20.5				
EVA	2	0.571	47.6	47.7	4.7				
SLASHDOT	3	1.777	2.3	24.1	67.1	6.5			
LOANS	3	2.06	0.1	13.9	66.3	19.7			
TWITTER	4	2.569	0.1	≈ 1	44.4	51.2	3.4		
EMAIL-VIRGILI	4	2.729	0.1	2.3	32	55.7	9.9		
EMAIL-ENRON	2	0.906	23.4	62.6	14				
EMAIL-EU	2	0.002	99.8	0.1	0.1				
WIKITALK-CHINA	3	2.076	≈ 0	0.01	92.4	7.6			
CE-METABOLIC	3	1.982	0.2	7.5	86.1	6.2			
SC-PP1	3	0.981	32.4	41.6	21.5	4.5			
YEAST-PP1	3	1.872	2	25.4	55.8	16.8			
HOMO-PI	2	0.747	34.2	56.9	8.9				
AS-GRAPH-1	3	1.791	0.5	24.9	69.7	4.9			
AS-GRAPH-2	3	1.124	9.6	68.5	21.7	0.2			
AS-GRAPH-3	2	0.828	27.8	61.6	10.6				
ROUTEVIEW	2	0.329	69.7	27.6	2.7				
AS-CAIDA	0	0	100						
ITDK	4	2.108	0.3	12	64.5	22.8	0.4		
GNUTELLA-06	4	2.507	0.3	5.7	41.1	48.8	4.1		
GNUTELLA-24	5	2.697	0.2	1.5	37	50.7	10.5	0.1	
GNUTELLA-30	5	3.167	0.1	1.8	13	52.4	31.8	0.9	
GNUTELLA-31	6	4.176	0.01	0.2	1.4	13.3	51.1	33.4	0.5
WEB-STANFORD	28	0.006	99.9	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
WEB-NOTREDAM	2	0.935	7.1	92.4	0.5				
WEB-BERKSTAN	22	0.002	99.9	≈ 0	0				
AMAZON-1	6	0.991	28.1	48	21.5	1.5	0.5	≈ 0	0.1
AMAZON-2	6	3.735	0.1	0.3	3.6	33.9	46.5	15.3	0.3
ROAD-EURO	8	0.135	97.4	0.3	0.1	0.4	0	0.8	≈ 1
OPENFLIGHT	3	1.879	0.2	23.9	63.7	12.2			
POWER-GRID	13	5.735	14.3	13.1	1.6	1.6	3.9	8.7	39.8
ROAD-PA	98	23.339	0.02	1.5	0.1	2.9	0.2	0.2	95

Table 8: Distribution of values $k(x) = ecc_T(x) - ecc_G(x)$, $x \in V$. $k_{max} := \max_{x \in V} k(x)$. $k_{avg} := \frac{1}{n} \sum_{x \in V} k(x)$.

Network	k_{max}	k_{avg}	% of vertices with $k(x) = 0$	% of vertices with $k(x) = 1$	% of vertices with $k(x) = 2$	% of vertices with $k(x) = 3$	% of vertices with $k(x) = 4$	% of vertices with $k(x) = 5$	% of vertices with $k(x) \geq 6$
DUTCH-ELITE	6	2.431	16.1	0	47.1	0	35.9	0	0.8
FACEBOOK	3	0.704	43.6	42.5	13.8	0.1			
EVA	2	0.572	47.6	47.6	4.8				
SLASHDOT	3	1.88	0.1	17.7	76.2	≈ 6			
LOANS	3	2.031	0.1	14	68.7	17.2			
TWITTER	3	1.821	3.1	≈ 20	68.6	8.3			
EMAIL-VIRGILI	4	1.932	4.3	22.8	48.4	24.4	0.1		
EMAIL-ENRON	2	0.903	22.4	64.8	12.7				
EMAIL-EU	2	0.002	99.9	0.03	0.1				
WIKITALK-CHINA	3	1.791	≈ 0	21	79	0.008			
CE-METABOLIC	1	0.349	65.1	34.9					
SC-PPI	7	4.196	1.3	4.1	6.2	13.4	27.2	35.9	11.8
YEAST-PPI	4	2.558	0.7	5.9	≈ 36	51.7	5.7		
HOMO-PI	2	0.612	41.6	55.5	2.9				
AS-GRAPH-1	2	0.887	19.6	72.2	8.2				
AS-GRAPH-2	2	0.833	25.7	65.3	≈ 9				
AS-GRAPH-3	2	0.312	70.4	28	1.6				
ROUTEVIEW	2	0.329	69.7	27.6	2.7				
AS-CAIDA	0	0	100						
ITDK	5	2.702	0.3	3.4	28.6	61.4	6.3	≈ 0	
GNUTELLA-06	5	3.543	0.01	0.7	5.9	37.2	50.9	5.3	
GNUTELLA-24	6	4.475	0.02	0.1	0.7	8.6	38.3	46.5	5.7
GNUTELLA-30	6	4.034	0.02	0.2	2.6	16.4	54.8	25.1	0.5
GNUTELLA-31	6	4.251	0.01	0.1	1.3	11.6	48.4	37.8	0.9
WEB-STANFORD	28	0.006	99.9	≈ 0	≈ 0	0.04	≈ 0	0.02	≈ 0
WEB-NOTREDAM	2	0.935	7.1	92.3	0.6				
WEB-BERKSTAN	22	0.002	99.97	≈ 0	0	0.02	0	0.01	≈ 0
AMAZON-1	7	0.919	49.7	21.7	18.6	8.1	1.1	0.4	0.3
AMAZON-2	5	2.03	1.2	15.1	65	17.1	1.6	≈ 0	
ROAD-EURO	8	0.135	97.4	0.3	0.1	0.4	0	0.8	≈ 1
OPENFLIGHT	2	0.641	36.1	63.7	0.2				
POWER-GRID	4	1.409	46.3	13.1	12.6	9.1	18.8		
ROAD-PA	80	22.545	0.7	20.9	0.3	0.2	0.4	0.2	77.3

Table 9: Distribution of values $k(x) = ecc_{T_2}(x) - ecc_G(x)$, $x \in V$. $k_{max} := \max_{x \in V} k(x)$. $k_{avg} := \frac{1}{n} \sum_{x \in V} k(x)$.