When Can Graph Hyperbolicity be Computed in Linear Time?*

Till Fluschnik[1,**], Christian Komusiewicz[2,***], George B. Mertzios[3], André Nichterlein[1,3,†], Rolf Niedermeier[1], and Nimrod Talmon[4,‡]

[1] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany,
{till.fluschnik, andre.nichterlein, rolf.niedermeier}@tu-berlin.de
[2] Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany,
christian.komusiewicz@uni-jena.de
[3] School of Engineering and Computing Sciences, Durham University, UK,
george.mertzios@durham.ac.uk
[4] Weizmann Institute of Science, Rehovot, Israel,
nimrodtalmon77@gmail.com

Abstract. Hyperbolicity measures, in terms of (distance) metrics, how close a given graph is to being a tree. Due to its relevance in modeling real-world networks, hyperbolicity has seen intensive research over the last years. Unfortunately, the best known practical algorithms for computing the hyperbolicity number of a n-vertex graph have running time $O(n^4)$. Exploiting the framework of parameterized complexity analysis, we explore possibilities for "linear-time FPT" algorithms to compute hyperbolicity. For instance, we show that hyperbolicity can be computed in time $2^{O(k)} + O(n + m)$ (m being the number of graph edges, k being the size of a vertex cover) while at the same time, unless the SETH fails, there is no $2^{o(k)}n^2$-time algorithm.

1 Introduction

(Gromov) hyperbolicity [16] of a graph is a popular attempt to capture and measure how *metrically* close a graph is to being a tree. The study of hyperbolicity is motivated by the fact that many real-world graphs are tree-like from a distance metric point of view [2, 3]. This is due to the fact that many of these graphs (including Internet application networks or social networks) possess certain geometric and topological characteristics. Hence, for many applications (cf., e.g. [3]), including the design of (more) efficient algorithms, it is useful to know

* This work was initiated at the yearly research retreat of the Algorithmics and Computational Complexity (AKT) group of TU Berlin, held in in Krölpa, Thuringia, Germany, from April 3rd till April 9th, 2016.
** Supported by the DFG, project DAMM (NI 369/13-2).
*** Supported by the DFG, project MAGZ (KO 3669/4-1).
† Supported by a postdoctoral fellowship of the DAAD while at Durham University.
‡ Supported by a postdoctoral fellowship from I-CORE ALGO.

the hyperbolicity of a graph. The hyperbolicity of a graph is a nonnegative number δ; the smaller δ is, the more tree-like the graph is; in particular, $\delta = 0$ means that the graph metric indeed is a tree metric. Typical hyperbolicity values for real-world graphs are below 5 [2].

Hyperbolicity can be defined via a four-point condition: Considering a size-four subset $\{a, b, c, d\}$ of the vertex set of a graph, one takes the (nonnegative) difference between the two largest of the three sums $\overline{ab} + \overline{cd}$, $\overline{ac} + \overline{bd}$, and $\overline{ad} + \overline{bc}$, where, e.g., $\overline{ab}$ denotes the length of the shortest path between vertices a and b in the given graph. The hyperbolicity is the maximum of these differences over all size-four subsets of the vertex set of the graph. For an n-vertex graph, this characterization of hyperbolicity directly implies a simple (brute-force) $O(n^4)$-time algorithm to compute its hyperbolicity. It has been observed that this running time is too slow for computing the hyperbolicity of large graphs as occurring in applications [2, 3, 4, 13]. On the theoretical side, it was shown that relying on some (rather impractical) matrix multiplication results, one can improve the upper bound to $O(n^{3.69})$ [13]. Moreover, roughly quadratic lower bounds are known [4, 13]. In practice, however, the best known algorithm still has an $O(n^4)$-time worst-case bound but uses several clever tricks when compared to the straightforward brute-force algorithm [3]. Indeed, based on empirical studies an $O(mn)$ running time is claimed, where m is the number of edges in the graph. Furthermore, there are heuristics for computing the hyperbolicity of a given graph [7].

To explore the possibility of faster algorithms for hyperbolicity in relevant special cases is the guiding principle of this work. More specifically, introducing some graph parameters, we investigate whether one can compute hyperbolicity in linear time when these parameters take small values. In other words, we employ the framework of parameterized complexity analysis (so far mainly used for studying NP-hard problems) applied to the polynomial-time solvable hyperbolicity problem. In this sense, we follow the recent trend of studying "FPT in P" [15]. Indeed, other than for NP-hard problems (where parameterized complexity is typically applied), for some parameters we achieve not only exponential dependence on the parameter but also polynomial ones. Note that such algorithms are unlikely for metric parameters like diameter or hyperbolicity.

Our contributions. Table 1 summarizes our main results. On the positive side, for a number of natural graph parameters we can attain "linear FPT" running times. Our "positive" graph parameters here are the following:

- the *covering path* number, that is, the minimum number of paths where only the endpoints have degree greater than two and which cover all vertices;
- the *feedback edge* number, that is, the minimum number of edges to delete to obtain a forest;
- the number of graph vertices of *degree at least three*;
- the *vertex cover* number, that is the minimum number of vertices needed to cover all edges in the graph;

Table 1. Summary of our algorithmic results. Herein, k denotes the parameter and n and m denote the number of vertices and edges, respectively.

Parameter	Running time	
covering path number	$O(k^4(n+m))$	[Theorem 5]
feedback edge number	$O(k^4(n+m))$	[Theorem 6]
number of $\geq$ 3-degree vertices	$O(k^8(n+m))$	[Theorem 8]
vertex cover number	$2^{O(k)} + O(n+m)$	[Theorem 10]
distance to cographs	$O(4^{4k} \cdot k^7 \cdot (n+m))$	[Theorem 15]

– the *distance to cographs*, that is, the minimum number of vertices to delete to obtain a cograph.[5]

On the negative side we prove that, with respect to the parameter vertex cover number k, we cannot hope for any $2^{o(k)}n^{2-\epsilon}$ algorithm unless the Strong Exponential Time Hypothesis (SETH) fails. We also obtain a "quadratic-time FPT" lower bound with respect to the parameter *maximum vertex degree*, again assuming SETH. Finally, we show that computing the hyperbolicity is at least as hard as computing a *size-four independent set* in a graph. It is conjectured that computing size-four independent sets needs $\Omega(n^3)$ time [20]. Due to lack of space, many details and proofs (marked with ($\star$)) had to be deferred.[6]

2 Preliminaries and Basic Observations

We write $[n] := \{1, \dots, n\}$ for every $n \in \mathbb{N}$. For a function $f : X \to Y$ and $X' \subseteq X$ we set $f(X') := \{y \in Y \mid \exists x \in X' : f(x) = y\}$.

Graph theory. Let $G = (V, E)$ be a graph. We define $|G| = |V| + |E|$. For $W \subseteq V$, we denote by $G[W]$ the graph *induced* by W. We use $G - W := G[V \setminus W]$ to denote the graph obtained from G by deleting the vertices of $W \subseteq V$. A *path* $P = (v_1, \dots, v_k)$ in G is a tuple of distinct vertices in V such that $\{v_i, v_{i+1}\} \in E$ for all $i \in [k-1]$; we say that such a path P has endpoints v_1 and v_k, we call the other vertices of P inner vertices, and we say that P is a v_1-v_k path. We denote by $\overline{ab}$ the length of a shortest a-b path if such a path exists; otherwise, that is, if a and b are in different connected components, we define $\overline{ab} := \infty$. Let $P = (v_1, \dots, v_k)$ be a path and v_i, v_j two vertices on P. We denote by $\overline{v_i v_j}|_P$ the distance of v_i to v_j on the path P, that is, $\overline{v_i v_j}|_P = |j - i|$. For a graph G we denote by $V_G^{\geq 3}$ the set of vertices of G that have degree at least three.

Hyperbolicity. Let $G = (V, E)$ be graph and $a, b, c, d \in V$. We define $D_1 := \overline{ab} + \overline{cd}$, $D_2 := \overline{ac} + \overline{bd}$, and $D_3 := \overline{ad} + \overline{bc}$ (referred to as *distance sums*). Moreover, we

[5] Cographs are the graphs without induced P_4s. Distance to cographs is upper-bounded by the parameter distance to cluster graph [10] and thus also by the parameter vertex cover number.

[6] A full version is available at https://arxiv.org/abs/1702.06503.

define $\delta(a,b,c,d) := |D_i - D_j|$ if $D_k \leq \min\{D_i, D_j\}$, for pairwise distinct $i,j,k \in \{1,2,3\}$. If any two vertices of the quadruple $\{a,b,c,d\}$ are not connected, we set $\delta(a,b,c,d) = 0$.[7] The *hyperbolicity* of $G = (V,E)$ is defined as $\delta(G) := \max_{a,b,c,d \in V}\{\delta(a,b,c,d)\}$. Note that by our definition, if G is not connected, $\delta(G)$ computes the maximal hyperbolicity over all connected components of G. We say that the graph is *δ-hyperbolic* for some $\delta \in \mathbb{N}$ if it has hyperbolicity at most δ. That is, a graph is δ-hyperbolic[8]. if for each 4-tuple $a,b,c,d \in V$ we have

$$\overline{ab} + \overline{cd} \leq \max\{\overline{ac} + \overline{bd}, \overline{ad} + \overline{bc}\} + \delta.$$

Formally, the Hyperbolicity problem is defined as follows.

Hyperbolicity
Input: An undirected graph $G = (V,E)$ and a positive integer δ.
Question: Is G δ-hyperbolic?

The following lemma will be useful later. For any quadruple $\{a,b,c,d\}$, Lemma 1 upper bounds $\delta(a,b,c,d)$ by twice the distance between any pair of vertices of the quadruple.

Lemma 1 ([7, Lemma 3.1]). $\delta(a,b,c,d) \leq 2 \cdot \min_{u \neq v \in \{a,b,c,d\}}\{\overline{uv}\}$

Reduction Rule 1. As long as there are more than four vertices, remove vertices of degree one.

Lemma 2 ($\star$). *Reduction Rule 1 is correct and can be exhaustively applied in linear time.*

3 Polynomial Linear-Time Parameterized Algorithms

In this section, we provide *polynomial linear-time parameterized* algorithms with respect to the parameters feedback edge number and number of vertices with degree at least three; that is, we present algorithms with running time having a linear-time dependence on the input size times a polynomial-time dependence on the parameter value (to which we refer to as PL-FPT running time).

To this end, we first introduce an auxiliary parameter, the *minimum maximal path cover number*, which we formally define below and also describe a polynomial linear-time parameterized algorithm for it.

Building upon this result, for the parameter feedback edge number we then show that, after applying Reduction Rule 1, the number of maximal paths can be upper-bounded by a polynomial of the feedback edge number. This implies a polynomial linear-time parameterized algorithm for the feedback edge number as well. For the parameter number of vertices with degree at least three, we

[7] This case is often left undefined in the literature. Our definition however enables to consider also disconnected graphs.

[8] Note that there is also a slightly different definition where graphs we call δ-hyperbolic are called 2δ-hyperbolic [7, 17]; we follow the definition of Brinkmann et al. [6].

introduce an additional reduction rule to achieve that the number of maximal paths is upper-bounded in a polynomial of this parameter. Again, this implies an algorithm with PL-FPT running time.

Minimum maximal path cover number.

Definition 3 (Maximal path). *Let G be a graph and P be a path in G. Then, P is a* maximal path *if the following holds: (1) P contains at least two vertices; (2) all its inner vertices have degree two in G; (3) either both its endpoints have degree at least three in G, or one of its endpoints has degree at least three in G while the other endpoint is of degree two in G; and (4) P is size-wise maximal with respect to these properties.*

We will be interested in the minimum number of maximal paths needed to cover the vertices of a given graph; we call this number the *minimum maximal path cover number*. While not all graphs can be covered by maximal paths (e.g., edgeless graphs), graphs which have minimum degree two and contain no isolated cycles, i.e. components that form induced cycles, can be covered by maximal paths (this follows by, e.g., a greedy algorithm which iteratively starts a path with an arbitrary uncovered vertex and exhaustively extends it arbitrarily; since there are no isolated cycles and the minimum degree is two, we are bound to eventually hit at least one vertex of degree three). Based on the approximation algorithm given in the next lemma, we assume in the following that we are given a maximal path cover.

Lemma 4 ($\star$). *There is a linear-time 2-approximation algorithm for the minimum maximal path cover number for graphs which have minimum degree two and contain no isolated cycles.*

Now we are ready to design a polynomial linear-time parameterized algorithm for Hyperbolicity with respect to the minimum maximal path cover number.

Theorem 5 ($\star$). *Let $G = (V, E)$ be a graph and k be its minimum maximal path cover number. Then,* Hyperbolicity *can be solved in $O(k^4(n+m))$ time.*

Feedback edge number. We next present a polynomial linear-time parameterized algorithm with respect to the parameter feedback edge number k. The idea is to show that a graph that is reduced with respect to Reduction Rule 1 contains $O(k)$ maximal paths.

Theorem 6 ($\star$). Hyperbolicity *can be computed in $O(k^4(n + m))$ time, where k is the feedback edge number.*

Number of vertices with degree at least three. We finally show a polynomial linear-time parameterized algorithm with respect to the number k of vertices with degree three or more. To this end, we use the following data reduction rule additionally to Reduction Rule 1 to bound the number of maximal paths in the graph by $O(k^2)$ (in order to make use of Theorem 5).

Reduction Rule 2. Let $G = (V, E)$ be a graph, $u, v \in V_G^{\geq 3}$ be two vertices of degree at least three, and $\mathcal{P}_{uv}$ be the set of maximal paths in G with endpoints u and v. Let $\mathcal{P}_{uv}^9 \subseteq \mathcal{P}_{uv}$ be the set containing the shortest path, the four longest even-length paths, and the four longest odd-length paths in $\mathcal{P}_{uv}$. If $\mathcal{P}_{uv} \setminus \mathcal{P}_{uv}^9 \neq \emptyset$, then delete in G all inner vertices of the paths in $\mathcal{P}_{uv} \setminus \mathcal{P}_{uv}^9$.

Lemma 7 ($\star$). *Reduction Rule 2 is correct and can be exhaustively applied in linear time.*

Observe that if the graph G is reduced with respect to Reduction Rule 2 after Reduction Rule 1 was applied, then for each pair $u, v \in V_G^{\geq 3}$ there exist at most nine maximal paths with endpoints u and v. Thus, G contains at most $O(k^2)$ maximal paths and using Theorem 5 we arrive at the following.

Theorem 8. Hyperbolicity *can be solved in $O(k^8(n + m))$ time, where k is the number of vertices with degree at least three.*

4 Parameter Vertex Cover

A *vertex cover* of a graph $G = (V, E)$ is a subset $W \subseteq V$ of vertices of G such that each edge in G is incident to at least one vertex in W. Deciding whether a graph G has a vertex cover of size at most k is NP-complete in general [14]. There is, however, a simple linear-time factor-2 approximation (see, e.g., [18]). In this section, we consider the size k of a vertex cover as the parameter. We show that we can solve Hyperbolicity in time linear in $|G|$, but exponential in k; further, we show that, unless SETH fails, we cannot do asymptotically better.

A Linear-Time Algorithm Parameterized by the Vertex Cover Number. We prove that Hyperbolicity can be solved in time linear in the size of the graph and exponential in the size k of a vertex cover. This result is based on a linear-time computable problem kernel of size $O(2^k)$ that can be obtained by exhaustively applying the following reduction rule.

Reduction Rule 3. If there are at least five vertices $v_1, v_2, \ldots, v_\ell \in V$, $\ell > 4$, with the same (open) neighborhood $N(v_1) = N(v_2) = \ldots = N(v_\ell)$, then delete $v_5, \ldots, v_\ell$.

We next show that the above rule is correct, can be applied in linear time, and leads to a problem kernel for the parameter vertex cover number.

Lemma 9 ($\star$). *Reduction Rule 3 is correct and can be applied exhaustively in linear time. Furthermore, if Reduction Rule 3 is not applicable, then the graph contains at most $k + 4 \cdot 2^k$ vertices and $O(k \cdot 2^k)$ edges, where k is the vertex cover number.*

With Reduction Rule 1 we can compute in linear time an equivalent instance having a bounded number of vertices. Applying to this instance the trivial $O(n^4)$-time algorithm yields the following.

Theorem 10. Hyperbolicity *can be computed in $O(2^{4k}+n+m)$ time, where k denotes the size of a vertex cover of the input graph.*

SETH-based Lower Bounds. We show that, unless SETH breaks, the $2^{O(k)} + O(n+m)$-time algorithm obtained in the previous subsection cannot be improved to an algorithm even with running time $2^{o(k)} \cdot n^{2-\epsilon}$. This also implies, that, assuming SETH, there is no problem kernel with $2^{o(k)}$ vertices computable in $O(n^{2-\epsilon})$ time, i. e., the kernel obtained by applying Reduction Rule 3 cannot be improved significantly. The proof follows by a many-one reduction from the problem Orthogonal Vectors: herein, given two sets $\overrightarrow{A}$ and $\overrightarrow{B}$ each containing n binary vectors of length $\ell = O(\log n)$, the question is whether there are two vectors $\overrightarrow{a} \in \overrightarrow{A}$ and $\overrightarrow{b} \in \overrightarrow{B}$ such that $\overrightarrow{a}$ and $\overrightarrow{b}$ are orthogonal, that is, such that there is no position i for which $\overrightarrow{a}[i] = \overrightarrow{b}[i] = 1$.

Williams and Yu [19] proved that, if Orthogonal Vectors can be solved in $O(n^{2-\epsilon})$ time, then SETH breaks. We provide a linear-time reduction from Orthogonal Vectors to Hyperbolicity where the graph G constructed in the reduction contains $O(n)$ vertices and admits a vertex cover of size $O(\log n)$ (and thus contains $O(n \cdot \log n)$ edges). The reduction then implies that, unless SETH breaks, there is no algorithm solving Hyperbolicity in time polynomial in the size of the vertex cover and linear in the size of the graph. We mention that Borassi et al. [4] showed that under the SETH Hyperbolicity cannot be solved in $O(n^{2-\epsilon})$. However, the instances constructed in their reduction have a minimum vertex cover of size $\Omega(n)$. Note that our reduction is based on ideas from the reduction of Abboud et al. [1] for the Diameter problem.

Theorem 11. *Assuming SETH,* Hyperbolicity *cannot be solved in* $2^{o(k)} \cdot (n^{2-\epsilon})$ *time, even on graphs with* $O(n \log n)$ *edges, diameter four, and domination number three. Here,* k *denotes the vertex cover number of the input graph.*

Proof. We reduce any instance $(\overrightarrow{A}, \overrightarrow{B})$ of Orthogonal Vectors to an instance (G, δ) of Hyperbolicity, where we construct the graph G as follows (we refer to Figure 1 for a sketch of the construction).

Make each $\overrightarrow{a} \in \overrightarrow{A}$ a vertex a and each $\overrightarrow{b} \in \overrightarrow{B}$ a vertex b of G, and denote these vertex sets by A and B, respectively. Add two vertices for each of the ℓ dimensions, that is, add the vertex set $C := \{c_1, \ldots, c_\ell\}$ and the vertex set $D = \{d_1, \ldots, d_\ell\}$ to G and make each of C and D a clique. Next, connect each $a \in A$ to the vertices of C in the natural way, that is, add an edge between a and c_i if and only if $\overrightarrow{a}[i] = 1$. Similarly, add an edge between $b \in B$ and $d_i \in D$ if and only if $\overrightarrow{b}[i] = 1$. Moreover, add the edge set $\{\{c_i, d_i\} \mid i \in [\ell]\}$. This part will constitute the central gadget of our construction.

Our aim is to ensure that the maximum hyperbolicity is reached for 4-tuples (a, b, c, d) such that $a \in A$, $b \in B$, and $\overrightarrow{a}$ and $\overrightarrow{b}$ are orthogonal vectors. The construction of G is completed by adding two paths (u_A, u, u_B) and (v_A, v, v_B), and making u_A and v_A adjacent to all vertices in $A \cup C$ and u_B and v_B adjacent to all vertices in $B \cup D$.

Observe that G contains $O(n)$ vertices, $O(n \cdot \log n)$ edges, and that the set $V \setminus (A \cup B)$ forms a vertex cover in G of size $O(\log n)$. Moreover, observe that G has diameter four. Note that each vertex in $A \cup B \cup C \cup D$ is at distance two to each

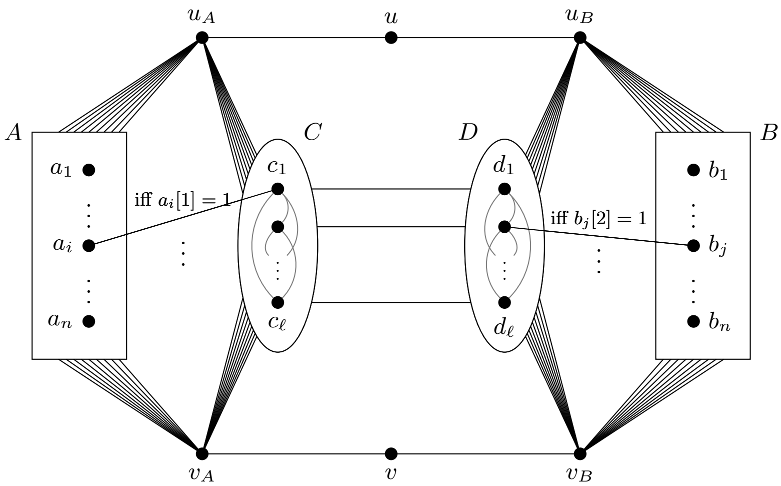

Fig. 1. Sketch of the construction described in the proof of Theorem 11. Ellipses indicate cliques, rectangles indicate independent sets. Multiple edges to an object indicate that the corresponding vertex is incident to each vertex enclosed within that object.

of u and v. Moreover, v_A and v_B are at distance three to u. Analogously, u_A, u_B are at distance three to v. Furthermore u and v are at distance four. Finally, observe that $\{u_A, u_B, v\}$ forms a dominating set in G.

We complete the proof by showing that $(\overrightarrow{A}, \overrightarrow{B})$ is a yes-instance of Orthogonal Vectors if and only if G has hyperbolicity at least $\delta = 4$.

($\Rightarrow$) Let $(\overrightarrow{A}, \overrightarrow{B})$ be a yes-instance, and let $\overrightarrow{a} \in \overrightarrow{A}$ and $\overrightarrow{b} \in \overrightarrow{B}$ be a pair of orthogonal vectors. We claim that $\delta(a, b, u, v) = 4$. Since $\overrightarrow{a}$ and $\overrightarrow{b}$ are orthogonal, there is no $i \in [\ell]$ with $\overrightarrow{a}[i] = \overrightarrow{b}[i] = 1$ and, hence, there is no path connecting a and b only containing two vertices in $C \cup D$, and it holds that $\overline{ab} = 4$. Moreover, we know that $\overline{uv} = 4$ as that $\overline{au} = \overline{bu} = \overline{av} = \overline{av} = 2$. Thus, $\delta(a, b, u, v) = 8 - 4 = 4$, and G is 4-hyperbolic.

($\Leftarrow$) Let $S = \{a, b, c, d\}$ be a set of vertices such that $\delta(a, b, c, d) \geq 4$. By Lemma 1, it follows that no two vertices of S are adjacent. Hence, we assume without loss of generality that $\overline{ab} = \overline{cd} = 4$. Observe that all vertices of C and D have distance at most three to all other vertices. Similarly, each vertex of $\{u_A, v_A, u_B, v_B\}$ has distance at most three to all other vertices. (Consider for example u_A. By construction, u_A is a neighbor of all vertices in $A \cup C \cup \{u\}$ and, hence, u_A has distance at most two to v_A and to all vertices in D. Thus, u_A has distance at most three to v, B, u_B and v_B and therefore to all vertices of G. The arguments for v_A, u_B, and v_B are symmetric.)

It follows that $S \subseteq A \cup B \cup \{u, v\}$, and therefore at least two vertices in S are from $A \cup B$. Thus, assume without loss of generality that a is contained in A. By the previous assumption, we have that $\overline{ab} = 4$. This implies that $b \in B$ and $\overrightarrow{a}$ and $\overrightarrow{b}$ are orthogonal vectors, as every other vertex in $V \setminus B$ is at distance three to a and each $b' \in B$ with $\overrightarrow{b'}$ being non-orthogonal to $\overrightarrow{a}$ is at distance three to a. Hence, $(\overrightarrow{A}, \overrightarrow{B})$ is a yes-instance. □

We remark that, with the above reduction, the hardness also holds for the variants in which we fix one vertex (u) or two vertices (u and v). The reduction also shows that approximating the hyperbolicity of a graph within a factor of $4/3 - \epsilon$ cannot be done in strongly subquadratic time or with a PL-FPT running time with respect to the vertex cover number.

Next, we adapt the above reduction to obtain the following hardness result on graphs of bounded maximum degree.

Theorem 12 ($\star$). *Assuming SETH,* Hyperbolicity *cannot be solved in $f(\Delta)\cdot(n^{2-\epsilon})$ time, where Δ denotes the maximum degree of the input graph.*

5 Parameter Distance to Cographs

In this section we describe a *linear-time parameterized* algorithm for Hyperbolicity parameterized by the vertex deletion distance k to cographs; that is, we present an algorithm with linear dependence on the input size but arbitrary dependence on the parameter (to which we refer to as L-FPT). A graph is a cograph if and only if it is P_4-free. Given a graph G we can determine in linear time whether it is a cograph and return an induced P_4 if this is not the case. This implies that in $O(k \cdot (m+n))$ time we can compute a set $X \subseteq V$ of size at most $4k$ such that $G - X$ is a cograph.

A further characterization is that a cograph can be obtained from graphs consisting of one single vertex via unions and joins [5].

- A *union* of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the graph $(V_1 \cup V_2, E_1 \cup E_2)$.
- A *join* of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the graph $(V_1 \cup V_2, E_1 \cup E_2 \cup \{\{v_1, v_2\} | v_1 \in V_1, v_2 \in V_2\})$.

The union of t graphs and the join of t graphs are defined by taking successive unions or joins, respectively, of the t graphs in an arbitrary order. Each cograph G can be associated with a rooted cotree T_G. The leaves of T_G are the vertices of V. Each internal node of T_G is labeled either as a union or join node. For node v in T_G, let $L(v)$ denote the leaves of the subtree rooted at v. For a union node v with children $u_1, \ldots, u_t$, the graph $G[L(v)]$ is the union of the graphs $G[L(u_i)]$, $1 \le i \le t$. For a join node v with children $u_1, \ldots, u_t$, the graph $G[L(v)]$ is the join of the graphs $G[L(u_i)]$, $1 \le i \le t$.

The cotree of a cograph can be computed in linear time [8]. In a subroutine in our algorithm for Hyperbolicity we need to solve the following variant of Subgraph Isomorphism.

Colored Induced Subgraph Isomorphism

Input: An undirected graph $G = (V, E)$ with a vertex-coloring $\gamma : V \to \mathbb{N}$ and an undirected graph $H = (W, F)$, where $|W| = k$, with a vertex-coloring $\chi : W \to \mathbb{N}$.

Question: Is there a vertex set $S \subseteq V$ such that there is an isomorphism f from $G[S]$ to H such that $\gamma(v) = \chi(f(v))$ for all $v \in S$?

Informally, the condition that $\gamma(v) = \chi(f(v))$ means that every vertex is mapped to a vertex of the same color. We say that such an isomorphism *respects the colorings*. As shown by Damaschke [9], Induced Subgraph Isomorphism on cographs is NP-complete. Since this is the special case of Colored Induced Subgraph Isomorphism where all vertices in G and H have the same color, Colored Induced Subgraph Isomorphism is also NP-complete (containment in NP is obvious). In the following, we show that on cographs Colored Induced Subgraph Isomorphism can be solved by an L-FPT algorithm when the parameter k is the order of H.

Lemma 13 (⋆). Colored Induced Subgraph Isomorphism *can be solved in $O(3^k(n+m))$ time in cographs.*

We now turn to the algorithm for Hyperbolicity on graphs that can be made into cographs by at most k vertex deletions. The final step is to reduce Hyperbolicity to the problem Distance-Constrained 4-Tuple: herein, given an undirected graph $G = (V, E)$ and six integers $d_{\{a,b\}}$, $d_{\{a,c\}}$, $d_{\{a,d\}}$, $d_{\{b,c\}}$, $d_{\{b,d\}}$, and $d_{\{c,d\}}$, the question is whether there is a set $S \subseteq V$ of four vertices and a bijection $f\colon S \to \{a, b, c, d\}$ such that for each $x, y \in S$ we have $\overline{xy} = d_{\{f(x),f(y)\}}$.

Lemma 14 (⋆). Distance-Constrained 4-Tuple *can be solved in $O(4^{4k} \cdot k \cdot (n+m))$ time if $G - X$ is a cograph for some $X \subseteq V$ of size k.*

We solve Hyperbolicity by creating $O(k^6)$ instances of Distance-Constrained 4-Tuple as shown below.

Theorem 15. Hyperbolicity *can be solved in $O(4^{4k} \cdot k^7 \cdot (n+m))$ time, where k is the vertex deletion distance of G to cographs.*

Proof. Let $G = (V, E)$ be the input graph and $X \subseteq V$, $|X| \le k$, such that $G - X$ is a cograph and observe that X can be computed in $O(4^k \cdot (n+m))$ time. Since every connected component of $G - X$ has diameter at most two, the maximum distance between any pair of vertices in the same component of G is at most $4k+2$: any shortest path between two vertices u and v visits at most k vertices in X, at most three vertices between every pair of vertices x and x' from X and at most three vertices before encountering the first vertex of X and at most three vertices before encountering the last vertex of X.

Consequently, for the 4-tuple (a, b, c, d) that maximizes $\delta(a, b, c, d)$, there are $O(k^6)$ possibilities for the pairwise distances between the four vertices. Thus, we may compute whether there is a 4-tuple such that $\delta(a, b, c, d) = \delta$ by checking for each of the $O(k^6)$ many 6-tuples of possible pairwise distances of four vertices in G whether there are 4 vertices in G with these six pairwise distances and whether this implies $\delta(a, b, c, d) \ge \delta$. The latter check can be performed in $O(1)$ time, and the first is equivalent to solving Distance-Constrained 4-Tuple which can be done in $O(4^{4k} \cdot k \cdot (n+m))$ time by Lemma 14. The overall running time follows. □

6 Reduction from 4-Independent Set

In this section, we provide a further relative lower bound for Hyperbolicity. Specifically, we prove that, if the running time is measured in terms of n, then Hyperbolicity is at least as hard as the problem of finding an independent set of size four in a graph. The currently best running time for this problem is $O(n^{3.257})$ [11, 20]. Hence, any improvement on the running time of Hyperbolicity which breaks this bound (e.g., an algorithm running in $o(n^3)$ time), would also yield a substantial improvement for the 4-Independent Set problem.

To this end, we reduce from a 4-partite (or 4-colored) variant of the Independent Set problem. The standard reduction [12] from Independent Set to Multicolored Independent Set shows that this 4-colored variant has the same asymptotic running time lower bound as 4-Independent Set.

Theorem 16 ($\star$). *Any algorithm solving* Hyperbolicity *in $O(n^c)$ time for some constant c yields an $O(n^c)$-time algorithm solving* 4-Independent Set.

7 Conclusion

To efficiently compute the hyperbolicity number, parameterization sometimes may help. In this respect, perhaps our practically most promising results relate to the $O(k^4(n+m))$ running times (for the parameters covering path number and feedback edge number, see Table 1). Note that they clearly improve on the standard algorithm when $k = O(n^{1/4})$. Moreover, the linear-time data reduction rules we presented may be of independent practical interest. On the lower bound side, together with the work of Abboud et al. [1] our SETH-based lower bound with respect to the parameter vertex cover number is among few known "exponential lower bounds" for a polynomial-time solvable problem.

As to future work, we particularly point to the following open questions. First, we left open whether there is an L-FPT algorithm exploiting the parameter feedback vertex number for computing the hyperbolicity number. Second, for parameter vertex cover number we have an SETH-based exponential lower bound for the parameter function in any L-FPT algorithm. This does not imply that it is impossible to achieve a polynomial parameter dependence when asking for algorithms with running time factors such as $O(n^2)$ or $O(n^3)$.

Bibliography

[1] A. Abboud, V. Vassilevska Williams, and J. R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proc. 27th SODA*, pages 377–391. SIAM, 2016.

[2] M. Abu-Ata and F. F. Dragan. Metric tree-like structures in real-world networks: an empirical study. *Networks*, 67(1):49–68, 2016.

[3] M. Borassi, D. Coudert, P. Crescenzi, and A. Marino. On computing the hyperbolicity of real-world graphs. In *Proc. 23rd ESA*, volume 9294 of *LNCS*, pages 215–226, 2015.

[4] 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.

[5] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: a Survey*, volume 3 of *SIAM Monographs on Discrete Mathematics and Applications*. SIAM, 1999.

[6] G. Brinkmann, J. H. Koolen, and V. Moulton. On the hyperbolicity of chordal graphs. *Annals of Combinatorics*, 5(1):61–69, 2001.

[7] N. Cohen, D. Coudert, and A. Lancin. On computing the Gromov hyperbolicity. *ACM Journal of Experimental Algorithmics*, 20:1.6:1–1.6:18, 2015.

[8] D. G. Corneil, Y. Perl, and L. K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985.

[9] P. Damaschke. Induced subgraph isomorphism for cographs is NP-complete. In *Proc. 16th WG*, volume 484 of *LNCS*, pages 72–78. Springer, 1991.

[10] M. Doucha and J. Kratochvíl. Cluster vertex deletion: A parameterization between vertex cover and clique-width. In *Proc. 37th MFCS*, volume 7464 of *LNCS*, pages 348–359. Springer, 2012.

[11] F. Eisenbrand and F. Grandoni. On the complexity of fixed parameter clique and dominating set. *Theoretical Computer Science*, 326(1-3):57–67, 2004.

[12] M. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.

[13] H. Fournier, A. Ismail, and A. Vigneron. Computing the Gromov hyperbolicity of a discrete metric space. *Information Processing Letters*, 115(6-8): 576–579, 2015.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[15] A. C. Giannopoulou, G. B. Mertzios, and R. Niedermeier. Polynomial fixed-parameter algorithms: A case study for longest path on interval graphs. In *Proc. 10th IPEC*, volume 43 of *LIPIcs*, pages 102–113. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[16] M. Gromov. Hyperbolic groups. In *Essays in Group Theory, MSRI Publ., vol. 8*, pages 75–263. Springer New York, 1987.

[17] D. Mitsche and P. Pralat. On the hyperbolicity of random graphs. *The Electronic Journal of Combinatorics*, 21(2):P2.39, 2014.

[18] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.

[19] R. Williams and H. Yu. Finding orthogonal vectors in discrete structures. In *Proc. 25th SODA*, pages 1867–1877. SIAM, 2014.

[20] V. V. Williams, J. R. Wang, R. Williams, and H. Yu. Finding four-node subgraphs in triangle time. In *Proc. 26th SODA*, pages 1671–1680. SIAM, 2015.