Proc. 16th ISAAC-05, Vol. 3827 in LNCS, pp. 1100-1109, Springer, 2005

Combinatorial Network Abstraction by Trees and Distances

Stefan Eckhardt[1], Sven Kosub[1], Moritz G. Maaß[1,⋆], Hanjo Täubig[1,⋆⋆], and Sebastian Wernicke[2,⋆⋆⋆]

[1] Fakultät für Informatik, Technische Universität München, Boltzmannstraße 3, D-85748 Garching, Germany
{eckhardt, kosub, maass, taeubig}@in.tum.de
[2] Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany
wernicke@minet.uni-jena.de.

Abstract. This work draws attention to combinatorial network abstraction problems which are specified by a class $\mathcal{P}$ of pattern graphs and a real-valued similarity measure ϱ based on certain graph properties. For fixed $\mathcal{P}$ and ϱ, the optimization task on any graph G is to find a subgraph G' which belongs to $\mathcal{P}$ and minimizes $\varrho(G, G')$. We consider this problem for the natural case of trees and distance-based similarity measures. In particular, we systematically study spanning trees of graphs that minimize distances, approximate distances, and approximate closeness-centrality with respect to some standard vector and matrix norms. The complexity analysis shows that all considered variants of the problem are NP-complete, except for the case of distance-minimization with respect to the L_∞ norm. We further show that unless P = NP, there exist no polynomial-time constant-factor approximation algorithms for the distance-approximation problems if a subset of edges can be forced into the spanning tree.

1 Introduction

Motivation. Network analysis aims at algorithmically exposing certain meaningful structures and characteristics of a complex network that can be considered essential for its functionality (see, e.g., [3] for a recent survey). A (simple) subnetwork containing only the essential parts of a given network is what we refer to as a *network abstraction.*

In this work, we formalize the combinatorial network abstraction problem by specifying a class $\mathcal{P}$ of admissible pattern graphs and a real-valued similarity measure ϱ that rates the degree of correct approximation of a given graph G by a subgraph $G' \subseteq G$ based on certain graph properties. For a fixed pattern class $\mathcal{P}$

⋆ Supported by DFG, grant Ma 870/5-1 (Leibnizpreis Ernst W. Mayr)
⋆⋆ Supported by DFG, grant Ma-870/6-1 (DFG-SPP 1126 Algorithmik großer und komplexer Netzwerke).
⋆⋆⋆ Supported by Deutsche Telekom Stiftung and Studienstiftung des deutschen Volkes.

and a fixed measure ϱ, the optimization task is to find for any input graph G a subgraph G' which belongs to $\mathcal{P}$ such that $\varrho(G, G')$ is minimal.

Here, we restrict ourselves to trees as the class of pattern graphs (although some results seem to easily carry over to related structures such as spanning subgraphs with a restricted number of edges) because they are the sparsest and simplest subgraphs that may connect all vertices of a network. Moreover, for several applications the use of spanning trees as an approximation of the network has some promising advantages:

1. *Understanding network dynamics.* A recent study [15] of communication kernels (which handle the majority of network traffic) shows that the organization of many complex networks is heavily influenced by their scale-free spanning trees.
2. *Guiding graph-layout for large networks.* We can use elegant tree-layout algorithms for drawing a tree that closely reflects the main characteristics of a given network.
3. *Compressing networks.* Even with most complex networks being sparse themselves, abstraction by trees reduces network sizes significantly.

In search of suitable graph properties for which a high amount of similarity between a network and its abstraction is desirable, we concentrate in this paper on distances as an inherent graph property. To quantify this degree of similarity, we use standard vector and matrix norms $\|\cdot\|_r$ (see Sect. 2 for a review and definitions) on the distance matrix D_G of an input graph G and the distance matrices of its spanning trees. To this end, we consider the following three optimization problems:

1. *Find a spanning tree that minimizes distances.* This corresponds to a similarity measure $\varrho_r(G, T) = \|D_T\|_r$. As an example, for the L_1 norm, the tree realizing the minimum is known as the minimum average distance tree (or, MAD-tree for short) [14, 8]. For the L_∞ matrix norm, the tree realizing the minimum is known as the minimum diameter spanning tree [6, 12].
2. *Find a spanning tree that approximates distances.* This corresponds to a similarity measure $\varrho_r(G, T) = \|D_T - D_G\|_r$. As an example, for the L_∞ matrix norm, we seek a tree that, for all vertex pairs, does not exceed a certain amount of additive increase in distance. Such trees are known as additive tree-spanners [17]. With the L_1 norm, we are again looking for a MAD-tree.
3. *Find a spanning tree that approximates centralities.* In this paper, we consider the popular notion of closeness centrality [2, 23] which, for any graph $G = (V, E)$ and vertex $v \in V$, is defined as $c_G(v) = (\sum_{t \in V} d_G(v, t))^{-1}$. The optimization problem is then based on the similarity measure $\varrho_r(G, T) = \|c_G - c_T\|_r$ for some vector norm $\|\cdot\|_r$.

Note that—except for the L_1 matrix norm—distance-minimizing spanning trees and optimal distance-approximating spanning trees typically cannot be used to

Proc. 16th ISAAC-05, Vol. 3827 in LNCS, pp. 1100-1109, Springer, 2005

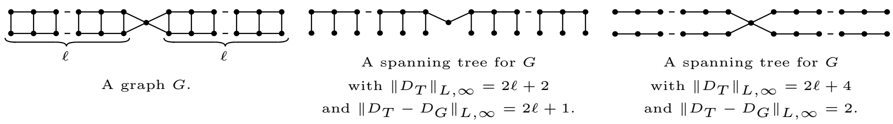

Fig. 1. Distance-minimization and distance-approximation do not provide good approximate solutions for each other with respect to the norm L_∞.

provide good approximate solutions for each other. An example for this (with respect to L_∞) is given in Fig. 1.

Results. We study the impact of the norm on the computational complexity of the above-mentioned network abstraction problems. For computing distance-minimizing spanning trees, two results have already been known, namely that there exists a polynomial-time algorithm for computing a minimum diameter spanning tree [6, 12] and that it is NP-complete to decide on input (G, γ) whether there is a spanning tree T of G such that $\|D_T\|_{L,1} \leq \gamma$ [14]. For distance-approximating spanning trees, even for L_1 and L_∞, no such results have so far been established to the best of our knowledge.[1]

In Sect. 3.2, we prove that deciding whether there exists a spanning tree T such that $\|D_T\|_r \leq \gamma$ for any given instance (G, γ) is NP-complete for all matrix norms within our framework where complexity has been unknown so far. We also consider fixed-edge versions (as, e.g., in [5]) where problem instances additionally specify a set of edges E_0 that must be contained in the spanning tree. If we allow arbitrary edge sets for E_0, then even Minimum Diameter Spanning Tree becomes NP-complete.

In Sect. 3.3, we prove that deciding whether there is a spanning tree T of G such that $\|D_T - D_G\| \leq \gamma$ for any given instance (G, γ) is NP-complete for all matrix norms within our framework, i.e., essentially for all standard norms (with exception of the spectral norm, a case which is left open). This is somewhat surprising, since at least in the case of L_∞ one might have hoped for a polynomial-time algorithm based on the polynomial-time algorithms for computing minimum diameter spanning trees. We also prove that the fixed-edge versions of finding optimal distance-approximating spanning trees cannot be approximated in polynomial-time within constant factor unless P = NP.

Finally, in Sect. 3.4, we prove that with respect to closeness centrality, deciding for a given instance (G, γ) whether there is a spanning tree T such that $\|c_G - c_T\|_r \leq \gamma$ is NP-complete for the L_1 vector norm.

Related work. Besides the already mentioned minimum diameter spanning trees [6, 12] and MAD-trees [14, 8], several notions of distance-approximability by trees have been considered in the literature. One variant is obtained by considering the *stretch* $d_T(u, v)/d_G(u, v)$ over all distinct vertices $u, v \in V$. If

[1] Note that in contrast to some claims in the literature the results in [18] do *not* provide a proof for the NP-completeness of deciding whether there is a spanning tree T with $\|D_T - D_G\|_{L,\infty} \leq \gamma$, neither does an easily conceivable adaption.

the stretch is at most γ, then the tree is called γ-multiplicative tree spanner (see, e.g., [22]—recently, also combinations of additive and multiplicative tree-spanners have been studied [10]). Finding a minimum maximum-stretch tree is NP-hard even for unweighted planar graphs [11] and cannot be approximated by a factor better than $(1+\sqrt{5})/2$ unless P = NP [19]. The problem of finding a minimum *average*-stretch tree is also NP-hard [14].

Spanning *subgraphs* (not only trees) with certain bounds on distance increases have been intensively studied since the pioneering work in [1, 21, 7]. The most general formulation of a spanner problem is the following [18]: A spanning subgraph H of G is an $f(x)$-spanner for G if and only if $d_H(u, v) \leq f(d_G(u, v))$ for all $u, v \in V(G)$. The computational problem then is to find an $f(x)$-spanner with the minimum number of edges, a problem somewhat dual to ours since it fixes a bound on the distance increase and tries to minimize the size of the subgraphs, whereas we fix the size of the subgraph and try to minimize the bounds. In a series of papers, the hardness of the spanner problems has been exhibited (see, e.g., [20, 5, 4, 16]). The version closest to our problem is to ask for a given graph G and two given parameters m, t if there exists an additive t-spanner for G with no more than m edges. This problem is NP-complete [18]. In the case that $m = n - 1$ is fixed, it becomes the problem of finding the best possible distance-approximating spanning tree with respect to $\|\cdot\|_{L,\infty}$. However, the corresponding NP-completeness proof for the general case relies heavily on the number of edges in the instance and hence a translation to an NP-completeness proof for the tree case is not obvious.

2 Notation

We consider simple, undirected, and unweighted graphs G with vertex set V and edge set E. For two vertices $v, w \in V$, the distance between v and w (i.e., the minimum number of edges in a path between u to v) in G is denoted by $d_G(v, w)$. The corresponding distance matrix is denoted by D_G. Clearly, D_G is symmetric with all entries being non-negative. Moreover, for any spanning tree T of a graph G, we have $D_T[i, j] \geq D_G[i, j]$ for all $v_i, v_j \in V$. We use the following well-known norms to evaluate a matrix A in $\mathbb{R}^{n\times n}$:

– The L_p norms $\|A\|_{L,p} \stackrel{\text{def}}{=} \left(\sum_{i=1}^{n}\sum_{j=1}^{n}|a_{i,j}|^p\right)^{1/p}$ for $1 \leq p < \infty$.[2]
– The L_∞ norm $\|A\|_{L,\infty} \stackrel{\text{def}}{=} \max_{i,j\in\{1,\ldots,n\}}|a_{i,j}|$.
– The maximum-column-sum norm $\|A\|_1 \stackrel{\text{def}}{=} \max_{j\in\{1,\ldots,n\}}\sum_{i=1}^{n}|a_{i,j}|$.
– The maximum-row-sum norm $\|A\|_\infty \stackrel{\text{def}}{=} \max_{i\in\{1,\ldots,n\}}\sum_{j=1}^{n}|a_{i,j}|$.

Trivially, for symmetric matrices we have $\|A\|_1 = \|A\|_\infty$. Therefore, we only consider the maximum-column-sum norm in our results to avoid redundancy.

[2] In the last part of the paper, we use L_p norms for vectors as well: for any $1 \leq p < \infty$ and vector $x \in \mathbb{R}^n$, define $\|x\|_p \stackrel{\text{def}}{=} (\sum_{i=1}^{n}|x_i|^p)^{1/p}$.

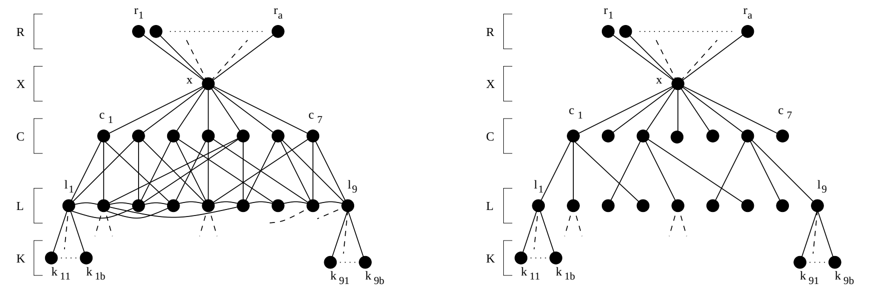

Fig. 2. Graph representation of an X3C instance and a corresponding solution tree.

3 Hardness Results

All our theorems establish hardness results that rely on similar constructions (which, however, depend on parameters that must be tuned in a non-trivial manner). We gather these essential constructions in the next section, followed by our results.

Note that, due to lack of space, we defer the proofs for our results to [9].

3.1 Gadgets

Graph representation of X3C instances. Given a family $\mathcal{C} = \{C_1, \ldots, C_s\}$ of 3-element subsets of a set $L = \{l_1, \ldots, l_{3m}\}$, the NP-complete problem Exact-3-Cover (X3C) asks whether there exists a subfamily $\mathcal{S} \subseteq \mathcal{C}$ of pairwise disjoint sets such that $\bigcup_{A \in \mathcal{S}} = L$. A subfamily $\mathcal{S}$ satisfying this property is called an *admissible solution* to $(\mathcal{C}, L)$. Suppose we are given an X3C instance $(\mathcal{C}, L)$ and let a, b be arbitrary natural numbers. Following a construction from [14], we define the graph $G_{a,b}(\mathcal{C}, L)$ to consist of the vertex set

$$V \overset{\text{def}}{=} \mathcal{C} \cup L \cup \underbrace{\{r_1, \ldots, r_a\}}_{\overset{\text{def}}{=} R} \cup \underbrace{\{x\}}_{\overset{\text{def}}{=} X} \cup \underbrace{\{k_{1,1}, \ldots, k_{1,b}, \ \ldots \ , k_{3m,1}, \ldots, k_{3m,b}\}}_{\overset{\text{def}}{=} K}$$

and the edge set

$$\begin{aligned} E \overset{\text{def}}{=} & \ \big\{ \ \{r_\mu, x\} \mid \mu \in \{1, \ldots, a\} \ \big\} \ \cup \ \big\{ \ \{C_\mu, x\} \mid \mu \in \{1, \ldots, s\} \ \big\} \ \cup \\ & \cup \big\{ \ \{l_\mu, C_\nu\} \mid l_\mu \in C_\nu \ \big\} \ \cup \ \big\{ \ \{l_\mu, l_\nu\} \mid \mu, \nu \in \{1, \ldots, 3m\} \ \big\} \ \cup \\ & \cup \big\{ \ \{k_{\mu,\nu}, l_\mu\} \mid \mu \in \{1, \ldots, 3m\} \text{ and } \nu \in \{1, \ldots, b\} \ \big\}. \end{aligned}$$

This construction is illustrated in Fig. 2. Given an admissible solution $\mathcal{S}$ to an X3C instance $(\mathcal{C}, L)$, we can identify a corresponding spanning subgraph $T_{\mathcal{S}}$ called *solution tree* in $G_{a,b}(\mathcal{C}, L)$ through the edge set

$$\begin{aligned} E(T_{\mathcal{S}}) = & \ \big\{ \ \{r_\mu, x\} \mid \mu \in \{1, \ldots, a\} \ \big\} \ \cup \ \big\{ \ \{C_\mu, x\} \mid \mu \in \{1, \ldots, s\} \ \big\} \ \cup \\ & \cup \big\{ \ \{l_\mu, C_\nu\} \mid l_\mu \in C_\nu \text{ and } C_\nu \in \mathcal{S} \ \big\} \ \cup \\ & \cup \big\{ \ \{k_{\mu,\nu}, l_\mu\} \mid \mu \in \{1, \ldots, 3m\} \text{ and } \nu \in \{1, \ldots, b\} \ \big\}. \end{aligned}$$

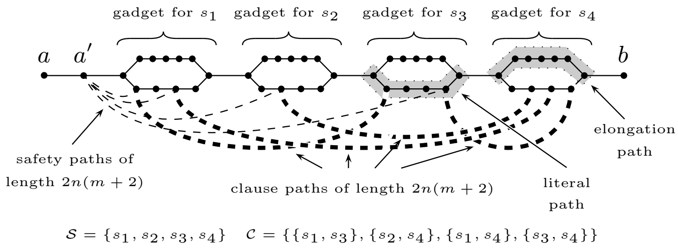

Fig. 3. Construction of a 2HS gadget $G(\mathcal{C}, \mathcal{S}, k)$. The dashed paths that are drawn bold consist solely of edges that must be contained in a spanning tree for the graph.

Lemma 1. *Let $(\mathcal{C}, L)$ be an* X3C *instance, $a, b \in \mathbb{N}$. Let T be any spanning tree of the graph $G_{a,b}(\mathcal{C}, L)$. There exists an admissible solution $\mathcal{S} \subseteq \mathcal{C}$ such that $T = T_{\mathcal{S}}$ if and only if the following conditions are satisfied:*

1. *For all $\mu \in \{1, \ldots, s\}$, the tree T contains the edge $\{C_\mu, x\}$.*
2. *For all $\mu \in \{1, \ldots, 3m\}$, there is a $\nu \in \{1, \ldots, s\}$ such that T contains the edge $\{l_\mu, C_\nu\}$.*
3. *For all $\mu \in \{1, \ldots, s\}$, vertex C_μ has either four neighbors in T or one.* □

Graph representation of 2HS instances. Given a family $\mathcal{C} = \{C_1, \ldots, C_m\}$ of 2-element subsets of a set $\mathcal{S} = \{s_1, \ldots, s_n\}$ and a natural number k, the NP-complete 2-Hitting Set (2HS) problem asks whether there exists a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $\|\mathcal{S}'\| \leq k$ and $\mathcal{S}' \cap C_\mu \neq \emptyset$ for all $\mu \in \{1, \ldots, m\}$.[3] A subset $\mathcal{S}' \subseteq S$ having this property is called an *admissible solution* to a 2HS instance $(\mathcal{C}, \mathcal{S}, k)$. Suppose we are given an instance $(\mathcal{C}, \mathcal{S}, k)$ of 2HS where $\|\mathcal{C}\| = m$ and $\|\mathcal{S}\| = n$. We define the graph $G(\mathcal{C}, \mathcal{S}, k)$ to consist of

- two vertices a, a', and b,
- for each $s_\mu \in \mathcal{S}$, consisting of vertices v_μ, v'_μ, $u^\mu_1, \ldots, u^\mu_{m+1}$ and $v^\mu_1, \ldots, v^\mu_m$,
- for each clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$, *clause paths* of length $2n(m+2)$ connecting v^ν_μ with v^κ_μ and *safety paths* of length $2n(m+2)$ connecting v^ν_μ with a'.

For each $s_\mu \in \mathcal{S}$ the *literal gadget* G_μ consists of two vertices v_μ and v'_μ called *connection vertices*. Both v_μ and v'_μ are connected via a path $(v_\mu, u^\mu_1, \ldots, u^\mu_{m+1} v'_\mu)$ of length $m+2$ called *elongation path* and a path $(v_\mu, v^\mu_1, \ldots, v^\mu_m v'_\mu)$ of length $m+1$ called the *literal path*. The construction is illustrated in Fig. 3.

Lemma 2. *Let $(\mathcal{C}, \mathcal{S}, k)$ be an instance of* 2HS. *Then we have $d_{G(\mathcal{C},\mathcal{S},k)}(a, b) = 2 + n(m+2)$. Moreover, there exists an admissible solution $\mathcal{S}' \subseteq \mathcal{S}$ to $(\mathcal{C}, \mathcal{S}, k)$ if and only if there exists a spanning tree T of $G(\mathcal{C}, \mathcal{S}, k)$ containing all edges in the clause paths such that $d_T(a, b) \leq d_{G(\mathcal{C},\mathcal{S},k)}(a, b) + k$.* □

[3] 2HS is better known as Vertex Cover. For the sake of readability (i.e., to avoid an overuse of the terms "vertices" and "edges"), we use the 2HS formulation.

3.2 Trees that Minimize Distances

Here we consider the problem of computing a spanning tree for a graph which minimizes distances among the vertices under certain matrix norms $\|\cdot\|_r$.

> *Problem:* Distance-Minimizing Spanning Tree (DMST)
> *Input:* A graph G and an algebraic number γ.
> *Question:* Does G contain a spanning tree T with $\|D_T\|_r \leq \gamma$?

Using the graph representation $G_{a,b}(\mathcal{C}, L)$ for any X3C instance $(\mathcal{C}, L)$, the following theorem can be shown.

Theorem 3. DMST *with respect to the norms* $\|\cdot\|_1$ *and* $\|\cdot\|_{L,p}$ *is* NP*-complete for all* $p \in \mathbb{N}_+$*, even when restricted to planar graphs.* □

It is known that a minimum-diameter spanning tree in a graph—i.e., DMST with respect to $\|\cdot\|_{L,\infty}$)—can be found in polynomial time [6, 12]. However, the next theorem shows that the *fixed-edge* version of this problem is intractable. This version additionally contains an edge set $E_0 \subseteq E(G)$ and we seek a spanning tree T such that $\|D_T\|_r \leq \gamma$ and $E_0 \subseteq E(T)$. Using the graph representation $G(\mathcal{C}, \mathcal{S}, k)$ for any given instance $(\mathcal{C}, \mathcal{S}, k)$ of 2HS gives us the following theorem.

Theorem 4. *The fixed-edge version of* DMST *with respect to the norm* $\|\cdot\|_{L,\infty}$ *is* NP*-complete.* □

3.3 Trees that Approximate Distances

We now turn to the problem of finding spanning trees that approximate the distances in a graph under a given matrix norm $\|\cdot\|_r$. We also examine the fixed-edge version of this problem, which is specified in the same way as for DMST.

> *Problem:* Distance-Approximating Spanning Tree (DAST)
> *Input:* A graph G and an algebraic number γ
> *Question:* Does G contain a spanning tree T with $\|D_T - D_G\|_r \leq \gamma$?

NP*-completeness results.* Using the graph representation $G_{a,b}(\mathcal{C}, L)$ for any X3C instance $(\mathcal{C}, L)$, the following theorem can be shown.

Theorem 5. DAST *with respect to the norms* $\|\cdot\|_1$ *and* $\|\cdot\|_{L,p}$ *is* NP*-complete for all* $p \in \mathbb{N}_+$. □

For proving the NP-completeness for the L_∞ matrix norm, it is helpful to first establish the result for the fixed-edge version (by reduction from 2HS).

Lemma 6. *The fixed-edge version of* DAST *with respect to the norm* $\|\cdot\|_{L,\infty}$ *is* NP*-complete.* □

To get rid of the fixed edges, we replace them by cycles such that deleting a fixed edge will cause the distance between two cycle vertices to increase by more than the allowed threshold γ, which then gives us the hardness result for the norm $\|\cdot\|_{L,\infty}$.[4]

Lemma 7. *Let $G = (V, E)$ be any graph and let $\{v, w\}$ be an arbitrary non-bridge edge in G. For $k > 3$, let G' be the graph resulting from adding a path $(v, u_1, \ldots, u_k, w)$ to G where $u_\mu \notin V$ for all $\mu \in \{1, \ldots, k\}$. There exists a spanning tree T of G which includes the edge $\{v, w\}$ and satisfies $\|D_T - D_G\|_{L,\infty} \leq k$ if and only if there exists a spanning tree T' of G' with $\|D_{T'} - D_{G'}\|_{L,\infty} \leq k$.* □

Theorem 8. DAST *with respect to the norm $\|\cdot\|_{L,\infty}$ is* NP*-complete.* □

Inapproximability results. We now show that—independent of the norm—it is hard to approximate the fixed-edge version of DAST in polynomial time within a constant factor. Let G be a graph and $E' \subseteq E(G)$ be the set of fixed edges. We say that an algorithm $\mathcal{A}$ is a *polynomial-time constant-factor approximation algorithm* for the fixed-edge version of DAST with respect to $\|\cdot\|_r$ if, for some constant $\delta > 0$, it computes a spanning tree $T_\mathcal{A}$ of G with $E' \subseteq E(T)$ in polynomial time such that $\|D_{T_\mathcal{A}} - D_G\|_r \leq \delta \cdot \|D_{T_{\text{opt}}} - D_G\|_r$. Here, T_{opt} is the *optimal tree*, i.e., $\|D_{T_{\text{opt}}} - D_G\|_r = \min_T \|D_T - D_G\|_r$ for all spanning trees T of G.

2HS is not polynomial-time approximable within a constant factor better than 7/6 unless P = NP [13]. To make use of this in our context, note that the main idea behind the graph representation $G(\mathcal{C}, \mathcal{S}, k)$ for any given instance $(\mathcal{C}, \mathcal{S}, k)$ of 2HS is that choosing an element from $\mathcal{S}$ into the solution corresponds to opening a literal path in $G(\mathcal{C}, \mathcal{S}, k)$. This opening is "penalized" by the elongation path, increasing the distance between the vertices a and b. The key idea now is to increase this penalty in a super-linear way by recursively replacing elongation paths with graph representations of the given 2HS instance.

More formally, suppose we are given an instance $(\mathcal{C}, \mathcal{S}, k)$ of 2HS where $\|\mathcal{C}\| = m$ and $\|\mathcal{S}\| = n$. For $j \in \mathbb{N}_+$ we define the graph $G(\mathcal{C}, \mathcal{S}, k, j)$ recursively as follows: For $j = 1$, the graph $G(\mathcal{C}, \mathcal{S}, k, j)$ is just the graph $G(\mathcal{C}, \mathcal{S}, k)$. For $j > 1$, define $l_{j-1} \stackrel{\text{def}}{=} d_{G(\mathcal{C}, \mathcal{S}, k, j-1)}(a, b)$. Then $G(\mathcal{C}, \mathcal{S}, k, j)$ consists of

- three vertices a, a', and b,
- *literal paths* P_μ for each $s_\mu \in \mathcal{S}$, consisting of vertices $v_1^\mu, \ldots, v_{l_{j-1}-1}^\mu$,
- *elongation gadgets* G_μ for each $s_\mu \in \mathcal{S}$, consisting of a copy of $G(\mathcal{C}, \mathcal{S}, k, j-1)$, with the vertices a, a', and b in $G(\mathcal{C}, \mathcal{S}, k, j-1)$ relabeled as a_μ, a'_μ and b_μ,
- for each $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$, *clause paths* and *safety paths* of length $2nl_{j-1}$ connecting v_μ^ν with v_μ^κ and v_μ^ν with a', respectively.

For each clause $C_\mu \in \mathcal{C}$ the vertices a_μ and b_μ are connected via a literal path $(a_\mu, v_1^\mu, \ldots, v_{l_{j-1}-1}^\mu, b_\mu)$. Furthermore, the edges $\{a, a'\}$, $\{a', a_1\}$, $\{b_m, b\}$, and the

[4] A similar technique with two cycles was used in [5, Lemma 3] to guarantee that any minimum t-spanner (i.e., a spanning subgraph with smallest number of edges such that $d_G(u, v) \leq t \cdot d_T(u, v)$ for all $u, v \in V$) contains a certain edge. However, this construction does not work in the context of additive distance growth and trees.

edges $\{b_i, a_{i+1}\}$ for all $1 \le i \le m-1$ are in $G(\mathcal{C}, \mathcal{S}, k, j)$. Note that the graph size is polynomial in the size of the instance $(\mathcal{C}, \mathcal{S}, k)$ and j. The following analogue to Lemma 2 can be established for our new graph representation of 2HS:

Lemma 9. *Let $(\mathcal{C}, \mathcal{S}, k)$ be a given instance of* 2HS *and $j \in \mathbb{N}_+$. Then, we have that $d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) = 3\frac{n^{j+1}-1}{n-1} + n^j(m-1) - 1$. Moreover, there exists an admissible solution $\mathcal{S}' \subseteq \mathcal{S}$ to $(\mathcal{C}, \mathcal{S}, k)$ if and only if there exists a spanning tree T of $G(\mathcal{C}, \mathcal{S}, k, j)$ containing all edges in the clause paths of all instances $G(\mathcal{C}, \mathcal{S}, k, j')$, for $j' < j$, of which $G(\mathcal{C}, \mathcal{S}, k, j)$ is composed, including the clause paths in $G(\mathcal{C}, \mathcal{S}, k, j)$ such that $d_T(a, b) \le d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) + k^j$.* □

Lemma 10. *Unless* P = NP*, there is no polynomial-time algorithm $\mathcal{A}$ that, given a* 2HS *instance $(\mathcal{C}, \mathcal{S}, k)$ and parameter $j \in \mathbb{N}_+$, computes a spanning tree $T_{\mathcal{A}}$ of $G(\mathcal{C}, \mathcal{S}, k, j)$ such that $d_{T_{\mathcal{A}}}(a, b) - d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) \le \delta \cdot d_{T_{opt}}(a, b) - d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b)$ for any $\delta > 0$. Here, T_{opt} is a spanning tree of $G(\mathcal{C}, \mathcal{S}, k, j)$ such that $d_{T_{opt}}(a, b) - d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) = \min_T d_T(a, b) - d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b)$ where the minimum is taken over all spanning trees that include all clause paths.* □

Theorem 11. *Unless* P = NP*, there is no polynomial-time constant-factor approximation algorithm for the fixed-edge version of* DAST *with respect to the norms $\|\cdot\|_{L,\infty}$, $\|\cdot\|_{L,p}$, and $\|\cdot\|_1$.* □

3.4 Trees that Approximate Centralities

Closeness centrality $c_G : V \to \mathbb{R}$ for a graph $G = (V, E)$ is defined for all $v \in V$ as $c_G(v) \stackrel{\text{def}}{=} (\sum_{t \in V} d_G(v, t))^{-1}$ [2, 23]. Here we consider the problem of computing a spanning tree such that its centrality function is as close as possible to the centrality function of the original graph with respect to some vector norm $\|\cdot\|_r$.

Problem: Closeness-Approximating Spanning Tree (CAST)
Input: A graph G and an algebraic number γ
Question: Does G contain a spanning tree T with $\|c_G - c_T\|_r \le \gamma$?

By a reduction from our X3C gadget, we can show the following theorem.

Theorem 12. CAST *with respect to the norm $\|\cdot\|_1$ is* NP*-complete, even when restricted to planar graphs.* □

4 Conclusion

We have introduced the problem of combinatorial network abstraction and systematically studied it for the natural case of trees and distance-based similarity measures. This provides the first computational complexity study in this area, presented in a unifying framework.

As an interesting problem left open here, future research might consider the presented problems with respect to the spectral norm—in the light that NP-completeness appears with coarser norms and the value of the spectral norm is always smaller than that of the norms considered here, there might even be a chance for polynomial-time solvability.

References

1. B. Awerbuch. Complexity of network synchronization. *J. ACM*, 32(4):804–823, 1985.
2. M. A. Beauchamp. An improved index of centrality. *Behavioral Science*, 10:161–163, 1965.
3. U. Brandes and T. Erlebach, editors. *Network Analysis: Methodological Foundations*, volume 3418 of *LNCS*. Springer-Verlag, 2005.
4. U. Brandes and D. Handke. NP-completeness results for minimum planar spanners. *Discr. Math. & Theor. Comp. Sci.*, 3(1):1–10, 1998.
5. L. Cai. NP-completeness of minimum spanner problems. *Discr. Appl. Math.*, 48(2):187–194, 1994.
6. P. M. Camerini, G. Galbiati, and F. Maffioli. Complexity of spanning tree problems: Part I. *Europ. J. Oper. Res.*, 5(5):346–352, 1980.
7. L. P. Chew. There are planar graphs almost as good as the complete graph. *J. Comp. Sys. Sci.*, 39(2):205–219, 1989.
8. E. Dahlhaus, P. Dankelmann, W. Goddard, and H. C. Swart. MAD trees and distance-hereditary graphs. *Discr. Appl. Math.*, 131(1):151–167, 2003.
9. S. Eckhardt, S. Kosub, M. G. Maaß, H. Täubig, and S. Wernicke. Combinatorial network abstraction by trees and distances. Technical Report TUM-I0502, Technische Universität München, Institut für Informatik, 2005.
10. M. Elkin and D. Peleg. $(1+\epsilon,\beta)$-spanner constructions for general graphs. *SIAM J. Comp.*, 33(3):608–631, 2004.
11. S. P. Fekete and J. Kremer. Tree spanners in planar graphs. *Discr. Appl. Math.*, 108(1–2):85–103, 2001.
12. R. Hassin and A. Tamir. On the minimum diameter spanning tree problem. *Inform. Proc. Lett.*, 53(2):109–111, 1995.
13. J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.
14. D. S. Johnson, J. K. Lenstra, and A. H. G. Rinnooy Kan. The complexity of the network design problem. *Networks*, 8:279–285, 1978.
15. D.-H. Kim, J. D. Noh, and H. Jeong. Scale-free trees: The skeletons of complex networks. *Phys. Rev. E*, 70(046126), 2004.
16. G. Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.
17. D. Kratsch, H.-O. Le, H. Müller, E. Prisner, and D. Wagner. Additive tree spanners. *SIAM J. Discr. Math.*, 17(2):332–340, 2003.
18. A. L. Liestman and T. C. Shermer. Additive graph spanners. *Networks*, 23(4):343–363, 1993.
19. D. Peleg and E. Reshef. Low complexity variants of the arrow distributed directory. *J. Comp. Sys. Sci.*, 63(3):474–485, 2001.
20. D. Peleg and A. A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989.
21. D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comp.*, 18(4):740–747, 1989.
22. E. Prisner. Distance approximating spanning trees. In *Proc. STACS'97*, volume 1200 of *LNCS*, pages 499–510. Springer-Verlag, 1997.
23. G. Sabidussi. The centrality index of a graph. *Psychometrica*, 31:581–603, 1966.