T U M

I N S T I T U T F Ü R I N F O R M A T I K

Combinatorial Network Abstraction by Trees and Distances

Stefan Eckhardt Moritz G. Maaß Sven Kosub
Hanjo Täubig Sebastian Wernicke

TUM-I0502
Februar 05

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM-INFO-02-I0502-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

Combinatorial Network Abstraction by Trees and Distances

Stefan Eckhardt   *Sven Kosub*   *Moritz G. Maaß**   *Hanjo Täubig* †

Fakultät für Informatik, Technische Universität München,
Boltzmannstraße 3, D-85748 Garching, Germany
{ eckhardt | kosub | maass | taeubig }@in.tum.de

Sebastian Wernicke‡

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Germany
wernicke@minet.uni-jena.de

Abstract

We draw attention to network abstraction as a fundamental problem within network analysis and visualization. A combinatorial network abstraction problem is 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 finding a subgraph G' which belongs to $\mathcal{P}$ such that $\varrho(G, G')$ is minimal. In this work, we consider this problem for the natural case of trees (as the class of pattern graphs) and similarity-measures based on distances. In particular—with respect to the most standard vector and matrix norms—we systematically study sub-trees of graphs that minimize distances, approximate distances, and approximate closeness-centrality. Although these similarity measures lead to reasonable results, the complexity analysis of finding optimal trees is discouraging: we prove that all problems are NP-complete independent of the norms used, except for the case of minimizing distances with respect to the L_∞ matrix-norm which was already known to have a polynomial algorithm.

1 Introduction

Network analysis aims at algorithmically exposing meaningful structures and characteristics of complex networks (there is a vast amount of literature on this topic and we refer to [3] for a recent survey).

*Supported by DFG grant Ma 870/5-1 (Leibnizpreis Ernst W. Mayr).

†Supported by DFG grant Ma 870/5-1 (Leibnizpreis Ernst W. Mayr) and 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. Main work done while the author was with TU München.

These structures might explain network functionality and the inter-relationships between its members on several levels of aggregation. It emerges as a fundamental issue to decide exactly which aspects of the network are those that can be considered essential for its functionality. A (simple) sub-network containing only the essential parts of a given network is what we refer to as *network abstraction*.

In this work, we formalize the combinatorial network abstraction problem by specifying a class $\mathcal{P}$ of pattern graphs and a real-valued similarity measure ϱ based on certain graph properties. For a fixed pattern class $\mathcal{P}$ 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. 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 amount of edges). The reasons for considering this class are obvious: trees are highly convenient network structures due to their structural simplicity—e.g., connectedness, planarity, bipartiteness—and algorithmically exploitable advantages such as recursive constructibility and sparseness. Moreover, for several applications the use of spanning trees as an approximation of the network has some promising advantages:

- *Understanding network dynamics.* Many dynamical phenomena of complex networks such as traffic and information flows are hard to predict solely from local information (such as degree distributions). A recent study [18] of communication kernels which handle most of the traffic of a network shows that the organization of many complex networks is heavily influenced by their scale-free spanning trees (that maximize the total sum of betweenness centralities).

- *Guiding graph-layout for large networks.* Instead of drawing a large and misty network, we can use elegant tree-layout algorithms for first drawing a tree having approximately the same characteristics as the network, and then add some missing edges of the overall network, if necessary at all.

- *Compressing networks.* Even with the most complex networks being sparse themselves (typically, the average degree is between two and four), trees reduce network sizes to 25-50 percent. Without essentially changing the network characteristics, this is worthwhile given storage limitations and time requirements.

Motivated by these examples, we systematically investigate the potential of trees as network abstractions. In the light of the examples above, we notably focus on computational feasibility. In search of suitable graph properties, to begin with, we concentrate in this paper on distances as one fundamental and well-studied network characteristic.

Given a connected and undirected graph $G = (V, E)$, let D_G denote the distance matrix of G, i.e., $D_G[u, u] = 0$ and $D_G[u, v] = d_G(u, v)$ for all $u, v \in V$. The best possible degree of similarity between D_G and distance matrices of spanning trees is an inherent graph property.[1] To quantify this degree of similarity, we use standard matrix norms $\|\cdot\|_r$ (reviewed in Sect. 2) that might be taken over vectors as well as over matrices. We consider the following three optimization problems:

- *Find a spanning tree that minimizes distances with respect to* $\|\cdot\|_r$. This corresponds to a similarity measure $\varrho_r(G, T) = \|D_T\|_r$. For the L_1 matrix-norm, i.e., $\|A\|_1 = \sum_i \sum_j |a_{ij}|$, the tree realizing the minimum is known as the minimum average distance tree (or, MAD-tree for

[1] Clearly, for all graphs G and G' over the same set of labeled vertices, we have that $D_G = D_{G'}$ if and only if $G = G'$. Thus, in general we cannot hope for exact representations by spanning trees.

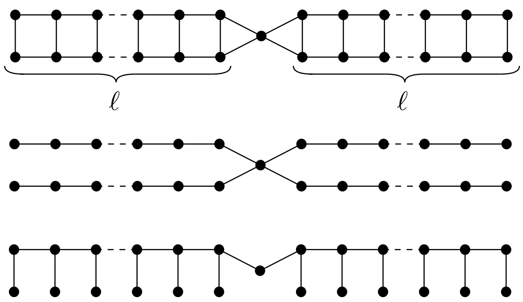

Figure 1: An illustration for the difference between distance minimization and distance approximation for the L_∞ matrix-norm.

short) [17, 10], and for the L_∞ matrix norm, i.e., $\|A\|_{L,\infty} = \max_{i,j} |a_{ij}|$, the tree realizing the minimum is known as the minimum diameter spanning tree [8, 16].

- *Find a spanning tree that approximates distances with respect to* $\|\cdot\|_r$. This corresponds to a similarity measure $\varrho_r(G,T) = \|D_T - D_G\|_r$. If we use the L_∞ matrix-norm, then we search for a tree that guarantees for all vertex pairs a certain additive increase in distance. Trees with more balanced increases are favored. Such trees are called additive tree-spanners [21, 25]. On the other hand, with respect to the L_1 matrix-norm, we search for a tree minimizing the average increase of distances between vertex pairs. Here, even some large deviations for few vertex pairs are allowed. Note that this tree is simultaneously a MAD-tree.

- *Find a spanning tree that approximates centralities with respect to* $\|\cdot\|_r$. A centrality c_G for a graph G is a mapping from vertices of G to the real numbers. Thus, this optimization problem is based on a similarity measure $\varrho_r(G,T) = \|c_G - c_T\|_r$ for some vector norm $\|\cdot\|_r$. In this paper, we consider the popular notion of closeness centrality [2, 26] 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 norms that we use throughout the paper are all standard and are reviewed in Sect. 2.

Note that except for the L_1 matrix-norm, distance-minimizing spanning trees and optimal distance-approximating spanning trees typically cannot be used to provide good approximate solutions for each other since the different underlying matrices drastically affect the cost of an optimal solution as well as the structure of the corresponding spanning tree: an example for this, with respect to L_∞, is provided by Fig. 1. Whilst the upper spanning tree provides a minimum diameter spanning tree for G, it approximates an optimal distance-approximating spanning tree only by a factor of $\Theta(\ell) = \Theta(\|V\|)$. The lower spanning tree is an optimal distance-approximating spanning tree for G whilst being suboptimal with respect to minimizing $\|D_T\|_{L,\infty}$. Note that there is no spanning tree for G which provides an optimal solution to both problems.

Figure 2 gives another example that separates optimal solutions for distance approximation and distance minimization for L_p. For arbitrary k and $\ell = 2^p(2k+4)+1$, the distance-minimizing spanning tree is the single center tree, which is not optimal with respect to distance approximation, while the best possible distance-approximating spanning tree is a multi-center tree for some r with $0 \leq r \leq k$, which however, is not optimal for distance minimization. When increasing ℓ to $2(2k+3)^p+1$, the single center solution is optimal for both distance approximation and distance minimization. This shows that distance approximation, to a certain degree, prefers locally good solutions over globally

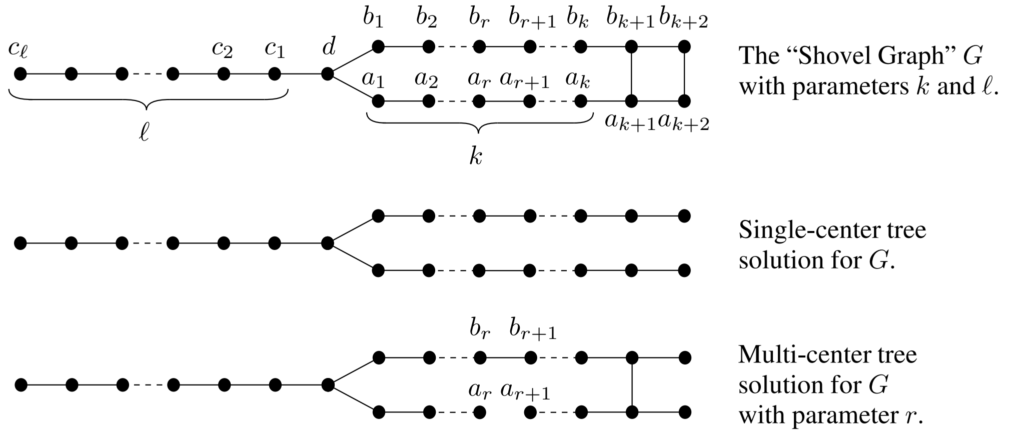

Figure 2: An illustration for the difference between distance minimization and distance approximation for L_p matrix-norms.

good solutions. When p increases, the "shaft" of the "shovel" can be made even longer compared to the "blade". Hence, a large p amplifies the local influence.

Results. In this paper, we study the complexity of the above-mentioned network abstraction problems. We are particularly interested in the impact of the norm on the complexity of the problems. For computing distance-minimizing spanning trees, we know that it is NP-complete to decide on input (G, γ) whether there is a spanning tree T of G such that $\|D_T\|_{L,1} \leq \gamma$ [17]. Moreover, we have a polynomial algorithm for computing a minimum diameter spanning tree [8, 16]. However, for computing distance-approximating spanning trees in general graphs, even for L_1 and L_∞, no such complexity results are known to the best of our knowledge.[2] Research in this area has more focused on proving the (non-)existence of certain distance-approximating trees for special graph classes (see, e.g., [25, 5, 13, 21]). These existence theorems are usually complemented with polynomial algorithms for finding the guaranteed trees. However, as our results show, we cannot break the curse of NP-completeness for our optimization problems:

- In Sect. 4, we prove that deciding whether there is 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 forced-edge versions (as, e.g., in [6, 15]), meaning that problem instances are of the form (G, E_0, γ) where E_0 consists of edges that must be contained in the spanning tree. If we allow arbitrary edge sets for E_0, then even the minimum diameter spanning tree problem becomes NP-complete.
- In Sect. 5, 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 the exceptional case of the spectral norm which is left open. This is somewhat surprising, since at least in the case of L_∞ (without forced edges), one might have hoped for a polynomial algorithm building up on the polynomial algorithms for computing minimum diameter spanning trees. Even worse, the polynomial al-

[2]Note that in contrast to some claim in the literature the results in [22] do *not* provide a proof for the NP-completeness of deciding whether there is a spanning tree T with $\|D_T - D_G\|_{L,\infty}(G) \leq \gamma$, neither does an easily conceivable adaption.

gorithm for computing the minimum diameter spanning tree cannot be used for approximating $\min_T \|D_T - D_G\|_{L,\infty}$ within reasonable factors. As the example in Fig. 1 shows, a minimum diameter spanning tree can be arbitrarily bad as a distance-approximating spanning tree of G.

- Finally, in Sect. 6 we prove that with respect to closeness centrality, deciding whether there is a spanning tree T such that $\|c_G - c_T\|_r \leq \gamma$ for any given instance (G, γ) is NP-complete for the L_1 vector-norm.

Related work. In addition to the already mentioned minimum diameter spanning trees [8, 16] and MAD-trees [17, 10], several notions of distance approximability by trees have been considered in the literature. One variant is obtained by relating pairs of vertices more closely: for a graph $G = (V, E)$ and a spanning tree T, we can consider the maximum of the ratio $d_T(u, v)/d_G(u, v)$ over all distinct vertices $u, v \in V$. If the maximum ratio is at most γ, then the tree is called γ-multiplicative tree spanner (see, e.g., [25]). As these notions are usually studied in a more general setting of spanning subgraphs, we do not take multiplicative tree-spanners into consideration. We should mention, however, that also combinations of additive and multiplicative tree-spanners have recently been proposed [11]. Another approach is based on (pseudo-)isometric trees [5, 21], where the minimization is not over spanning trees but over all trees having the same number of vertices as the network in question. Since this loses a direct linkage between the tree and the network, we do not follow this vein.

Spanning *subgraphs* (not only trees) with certain bounds on distance increases have been intensively studied since the pioneering work in [1, 24, 9]. These notions are typically motivated by problems in network design (see, e.g., [23, 7, 28, 15] and the surveys [27, 12]) and not (yet) in network analysis. The most general formulation of a spanner problem is the following [22]: a spanning subgraph H of G is an $f(x)$-spanner for G if and only if for all $u, v \in V(G)$, $d_H(u, v) \leq f(d_G(u, v))$. As examples, for $f(x) = t + x$ we obtain additive t-spanners, and for $f(x) = t \cdot x$ we obtain multiplicative t-spanners. The computational problem is to find an $f(x)$-spanner with the minimum number of edges. It is thus a problem dual to ours, as 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., [23, 6, 20, 19, 4]). The version closest to our problem is the following: given graph G and parameter m, it is asked for any fixed $k \geq 1$ if there is an additive t-spanner with no more than m edges. This problem has been proven to be NP-complete [22]. In case that $m = n - 1$ is fixed, the problem considered in [22] is just the decision version of finding the best possible distance-approximating spanning tree with respect to $\|\cdot\|_{L,\infty}$. However, the NP-completeness proof relies heavily on the number of edges in the instance. Thus, a translation to an NP-completeness proof for the tree case is not obvious. We resolve this issue here.

2 Notation

$\mathbb{N} = \{0, 1, 2, \ldots\}$ is the set of natural numbers, $\mathbb{N}_+ = \{1, 2, \ldots\}$. Throughout the paper we only consider simple, undirected, unweighted graphs. Let G be any such graph. Then $V(G)$ denotes the vertex set of G and $E(G)$ denotes the set of edges of G. If the graph is clear from the context, we simply use V and E. For vertices $v, w \in V(G)$, the distance between v and w in G, denoted by $d_G(v, w)$, is the minimum length of a path in G starting in v and ending in w. For a graph G with vertex set $\{v_1, \ldots, v_n\}$, we define $D_G \in \mathbb{N}^{n\times n}$ to be its distance matrix, i.e., for all $i, j \in \{1, \ldots, n\}$, the entries in D_G satisfy $D_G[i,j] = d_G(v_i, v_j)$. Clearly, D_G is a symmetric matrix with all entries being non-negative. Moreover, for any spanning tree T of a graph G, we have for all $v_i, v_j \in V$, $D_T[i,j] \geq D_G[i,j]$. We use the following well-known norms to evaluate a matrix A in $\mathbb{R}^{n \times n}$:

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

Trivially, for symmetric matrices we have $\|A\|_1 = \|A\|_\infty$. Thus, all our results regarding the maximum-column-sum norm also hold for maximum-row-sum norm and vice versa. Therefore, we only consider the maximum-column-sum norm. In the last part of the paper, we use L_p norms for vectors as well: for any vector $x \in \mathbb{R}^n$ define $\|x\|_p =_{\text{def}} \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$ for $1 \leq p < \infty$.

3 Gadgets

All our theorems establish NP-completeness results and the proofs all rely on similar constructions (which, however, depend on parameters that must be tuned in a non-trivial manner). We gather these essential constructions in this section.

Graph representation of X3C instances. Deciding whether there is an exact set-cover by sets having three elements each is the following, well-known NP-complete problem (see [14]).

Problem:	X3C
Input:	A family $\mathcal{C} = \{C_1, \ldots, C_s\}$ of 3-element subsets of a set $L = \{l_1, \ldots, l_{3m}\}$
Question:	Is there 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 $(\mathcal{C}, L)$. It is clear that $m \leq s$ and that any admissible solution consists of exactly m sets.

Suppose we are given an X3C instance $(\mathcal{C}, L)$. Let a and b be arbitrary natural numbers. Following a construction from [17], we define the graph $G_{a,b}(\mathcal{C}, L)$ to consist of the vertex set

$$V =_{\text{def}} \mathcal{C} \cup L \cup \underbrace{\{r_1, \ldots, r_a\}}_{=_{\text{def}} R} \cup \underbrace{\{x\}}_{=_{\text{def}} X} \cup \underbrace{\{k_{1,1}, \ldots, k_{1,b}, k_{2,1}, \ldots, k_{2,b}, \ldots, k_{3m,1}, \ldots, k_{3m,b}\}}_{=_{\text{def}} K}$$

and edge set

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

The representation of an X3C instance according to the graph definition is illustrated in Fig. 3. The following proposition summarizes some immediate observations.

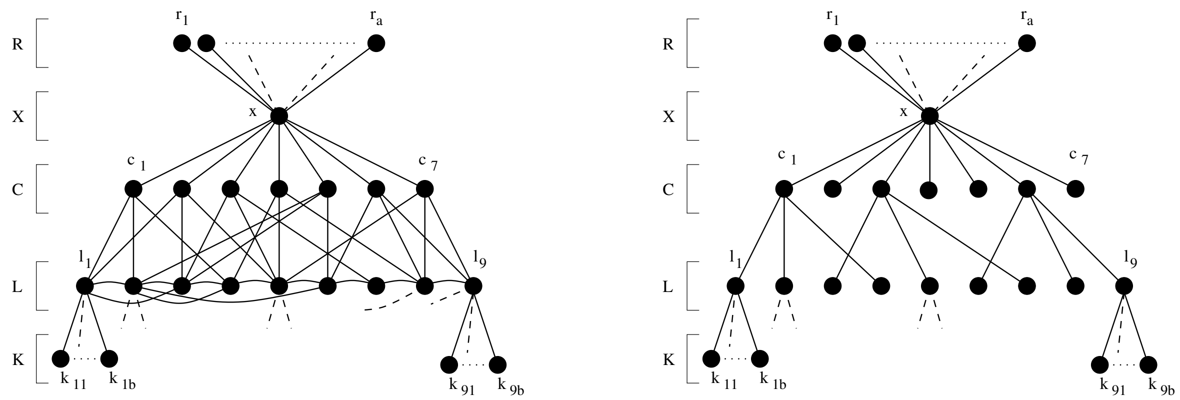

Figure 3: Graph representation of an X3C instance and a corresponding solution tree.

Proposition 1. *Let $(\mathcal{C}, L)$ be an* X3C *instance and let a and b be natural numbers. Suppose T is a spanning tree of the graph $G_{a,b}(\mathcal{C}, L)$.*

1. *For all $\mu \in \{1, \ldots, a\}$, T contains the edge $\{r_\mu, x\}$.*

2. *For all $\mu \in \{1, \ldots, 3m\}$ and all $\nu \in \{1, \ldots, b\}$, T contains the edge $\{k_{\mu,\nu}, l_\mu\}$.*

3. *If for some $\mu \in \{1, \ldots, s\}$, T does not contain the edge $\{C_\mu, x\}$, then for all $\nu \in \{1, \ldots, a\}$,*
$$d_T(C_\mu, r_\nu) \geq 4 \;\text{ and }\; d_T(C_\mu, r_\nu) \geq d_{G_{a,b}(\mathcal{C},L)}(C_\mu, r_\nu) + 2.$$

4. *If for some $\mu \in \{1, \ldots, 3m\}$, the vertex l_μ is not adjacent to any $C_\nu \in \mathcal{C}$, $\nu \in \{1, \ldots, s\}$, then for all $\kappa \in \{1, \ldots, a\}$ and $\lambda \in \{1, \ldots, b\}$,*
$$\begin{aligned} d_T(l_\mu, r_\kappa) \geq 4 \quad &\text{and} \quad d_T(k_{\mu,\lambda}, r_\kappa) \geq 5, \\ d_T(l_\mu, r_\kappa) \geq d_{G_{a,b}(\mathcal{C},L)}(l_\mu, r_\kappa) + 1 \quad &\text{and} \quad d_T(k_{\mu,\lambda}, r_\kappa) \geq d_{G_{a,b}(\mathcal{C},L)}(k_{\mu,\lambda}, r_\kappa) + 1. \end{aligned}$$

Assume that we are given an admissible solution $\mathcal{S}$ to an X3C instance $(\mathcal{C}, L)$. Then, we can identify a corresponding spanning subgraph $T_\mathcal{S}$ in the graph representation $G_{a,b}(\mathcal{C}, L)$ through the following edge set:

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

Since $\mathcal{S}$ consists of pairwise disjoint sets, $T_\mathcal{S}$ is a tree.

We observe some properties of solution trees.

Proposition 2. *Let $(\mathcal{C}, L)$ be an* X3C *instance having an admissible solution $\mathcal{S} \subseteq \mathcal{C}$. Let a and b be natural numbers. Let $T_\mathcal{S}$ be the spanning tree of $G_{a,b}(\mathcal{C}, L)$ that corresponds to $\mathcal{S}$.*

1. *For all $\mu \in \{1, \ldots, s\}$, if $C_\mu \in \mathcal{S}$, then C_μ has four neighbors in $T_\mathcal{S}$, otherwise C_μ has only one neighbor in $T_\mathcal{S}$.*

2. *For all vertices* $u \in R \cup X$ *and* $v \in V$, $d_{T_{\mathcal{S}}}(u, v) = d_{G_{a,b}(\mathcal{C},L)}(u, v)$.

3. *For all* $\mu, \nu \in \{1, \dots, s\}$, $d_{T_{\mathcal{S}}}(C_\mu, C_\nu) = d_{G_{a,b}(\mathcal{C},L)}(C_\mu, C_\nu)$.

The following lemma provides a structural characterization of spanning trees that correspond to admissible solutions.

Lemma 3. *Let* $(\mathcal{C}, L)$ *be an* X3C *instance. Let* a *and* b *be natural numbers. 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 all satisfied:*

1. *For all* $\mu \in \{1, \dots, s\}$, T *contains the edge* $\{C_\mu, x\}$.

2. *For all* $\mu \in \{1, \dots, 3m\}$, *there is a* $\nu \in \{1, \dots, s\}$ *such that* T *contains the edge* $\{l_\mu, C_\nu\}$.

3. *For all* $\mu \in \{1, \dots, s\}$, *the vertex* C_μ *has either four neighbors in* T *or one.*

Proof. Clearly, the three conditions are necessary for a tree $T_{\mathcal{S}}$ to correspond to an admissible solution $\mathcal{S}$. For the other direction, suppose the tree T satisfies all conditions. By the first and second conditions, for $\mu, \nu \in \{1, \dots, 3m\}$ such that $\mu \neq \nu$, there exist $\kappa, \lambda \in \{1, \dots, s\}$ such that the path $(l_\mu, \mathcal{C}_\kappa, x, \mathcal{C}_\lambda, l_\nu)$ exists in T. Thus edges $\{l_\mu, l_\nu\}$ do not belong to T. Consequently, using the third condition, we obtain an admissible solution by defining $\mathcal{S}$ to consist of all $\mathcal{C}_\mu$ having exactly four neighbors in T. □

Graph representation of 2-HITTING SET. 2-HITTING SET is better known as the NP-complete VERTEX COVER problem. However, we use the former problem formulation to avoid overuse of the terms "vertices" and "edges" for the sake of readability:

Problem:	2-HITTING SET
Input:	A family $\mathcal{C} = \{C_1, \dots, C_m\}$ of 2-element subsets of a set $\mathcal{S} = \{s_1, \dots, s_n\}$ and an integer k
Question:	Does there exists a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $\|\mathcal{S}'\| \leq k$ and for each $\mu \in \{1, \dots, m\}$, there is at least one element in $C_\mu \cap \mathcal{S}'$?

A subset $\mathcal{F} \subseteq S$ having this property is called an *admissible solution* to an 2-HITTING SET instance $(\mathcal{C}, \mathcal{F}, k)$.

Suppose we are given an instance $(\mathcal{C}, \mathcal{S}, k)$ of 2-HITTING SET where $\|\mathcal{C}\| = m$ and $\|\mathcal{S}\| = n$. We define the graph $G(\mathcal{C}, \mathcal{S}, k)$ to consists of

- vertices a, a', and b
- *literal gadgets* G_μ for each $s_\mu \in \mathcal{S}$, consisting of vertices $v_\mu, v'_\mu, u_1^\mu, \dots, u_{m+1}^\mu, v_1^\mu, \dots, v_m^\mu$.
- *clause paths* of length $2n(m+2)$ for each clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$ connecting v_μ^ν with v_μ^κ, and
- *safety paths* of length $2n(m+2)$ for each clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$, connecting v_μ^ν with a'.

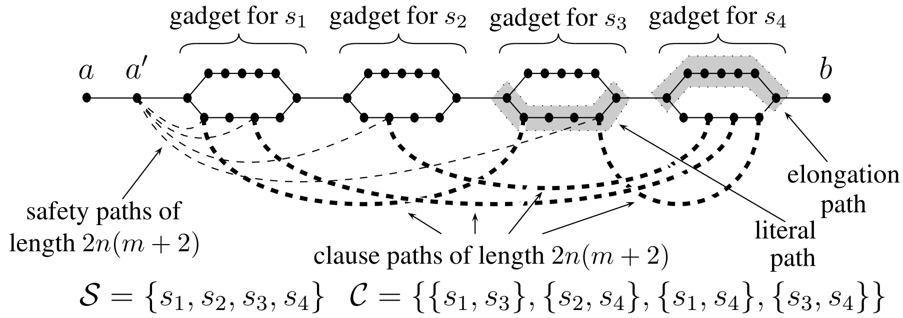

Figure 4: Construction of a Hitting Set Gadget $G(\mathcal{C}, \mathcal{S}, k)$ corresponding to a given instance of 2-HITTING SET. The dashed paths that are drawn bold consist solely of edges that must be contained in a spanning tree for the graph.

Note that the only purpose of the *clause paths* is capturing forced edges in some versions of our problems. 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 $(v_\mu, u_1^\mu, \ldots, u_{m+1}^\mu, v'_\mu)$ of length $m+2$ called *elongation path* and a path $(v_\mu, v_1^\mu, \ldots, v_m^\mu, v'_\mu)$ of length $m+1$ called the *literal path*. Clearly, the graph size is polynomial in the size of the instance $(\mathcal{C}, \mathcal{S}, k)$. The construction is illustrated in Fig. 4.

Lemma 4. *Let* $(\mathcal{C}, \mathcal{S}, k)$ *be an instance of* 2-HITTING SET.

1. *We have* $d_{G(\mathcal{C},\mathcal{S},k)}(a, b) = 2 + n(m+2)$.
2. *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* $d_T(a, b) \leq d_{G(\mathcal{C},\mathcal{S},k)}(a, b) + k$.

Proof. The first statement follows from the observation that any path from a to b using a clause or safety path has length at least $2 + 2n(m+2)$ while the shortest path between a and b via literal or elongation paths has length $n(m+1) + n + 2$.

For the second statement we prove the two directions separately.

($\Rightarrow$) Suppose $\mathcal{S}'$ is an admissible solution to $(\mathcal{C}, \mathcal{S}, k)$, i.e., $\|\mathcal{S}'\| \leq k$. Construct a spanning tree of $G(\mathcal{C}, \mathcal{S}, k)$ as follows:

1. For each clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$ do the following: if $s_\nu \in \mathcal{S}'$, then remove the edge $\{v_{\mu-1}^\nu, v_\mu^\nu\}$. If $s_\kappa \in \mathcal{S}'$, then remove the edge $\{v_{\mu-1}^\kappa, v_\mu^\kappa\}$. (We denoted $v_0^\nu = v_\nu$ and $v_0^\kappa = v_\kappa$ here.) If not both s_ν and s_κ are elements of $\mathcal{S}$, then remove an edge from the safety path between s_ν and a'.[3]
2. For each $s_\mu \in \mathcal{S}'$, remove edge $\{v_m^\mu, v'_\mu\}$.
3. For each $s_\mu \notin \mathcal{S}'$, remove the edge $\{v_\mu, u_1^\mu\}$.

Note that no edge from a clause path was removed. We ensured that each cycle induced by the literal and elongation paths is broken because at least one edge is removed by second and third construction rule. The cycles induced by the clause and safety paths are broken by the first and third construction rule: for each clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$ at least one of the sets $\{\{v_{\mu-1}^\nu, v_\mu^\nu\}, \{v_m^\nu, v'_\nu\}\}$

[3] Without the safety paths, the given edge removal scheme might leave a clause gadget disconnected in T.

and $\{\{v^\kappa_{\mu-1}, v^\kappa_\mu\}, \{v^\kappa_m, v'_\kappa\}\}$ is removed, thus either v^ν_μ or v^κ_μ is not reachable via the clause path from v_ν or v'_ν (v_κ or v'_κ, respectively). An edge from the safety path is removed, except if both $\{\{v^\nu_{\mu-1}, v^\nu_\mu\}, \{v^\nu_m, v'_\nu\}\}$ and $\{\{v^\kappa_{\mu-1}, v^\kappa_\mu\}, \{v^\kappa_m, v'_\kappa\}\}$ are removed, in which case neither v^ν_μ nor v^κ_μ is reachable via the clause path from any of $v_\nu, v'_\nu, v_\kappa, v'_\kappa$. A cycle induced by multiple clause paths not leading via any connection vertices cannot occur, since the connection is broken at one of the literals in $\mathcal{S}'$.

As a result, there is a path between a and b in T leading via elongation ($s_\mu \in \mathcal{S}'$) or literal ($s_\mu \notin \mathcal{S}'$) paths. By means of construction, the distance via a literal path is shorter by one than the distance via an elongation. Therefore,

$$d_T(a,b) = 2 + (n - \|\mathcal{S}\|)(m+2) + \|\mathcal{S}\|(m+3) = 2 + n(m+2) + \|\mathcal{S}\| \leq d_{G(\mathcal{C},\mathcal{S},k)}(a,b) + k.$$

($\Leftarrow$) Suppose T is a spanning tree of $G(\mathcal{C},\mathcal{S},k)$ containing all edges of the clause paths and satisfying $d_T(a,b) \leq d_{G(\mathcal{C},\mathcal{S},k)}(a,b) + k$. Since $d_{G(\mathcal{C},\mathcal{S},k)}(a,b) + k \leq 2 + n(m+3) < 2 + 2n(m+2)$, the path cannot lead via any clause or safety paths. Hence, it must lead via literal and elongation paths only. The length of any (intact) elongation path is $m+2$, the length of any (intact) literal path is $m+1$. Therefore, the path from a to b leads over exactly k elongation paths. Let $\mathcal{S}'$ be the set of literals s_μ for which the path leads from v_μ to v'_μ via an elongation path. Here, the literal path must be broken (due to the minimality of the path length). Conversely, for every $s_\mu \notin \mathcal{S}'$, the literal path is not broken, i.e., $(v_\mu, v^\mu_1, \dots, v^\mu_m, v'_\mu)$ is a path in T. Assume that we have for any clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$ (where $\nu < \kappa$), $C_\mu \cap \mathcal{S}' = \emptyset$. The clause path connects v^ν_μ with v^κ_μ. Since $s_\nu, s_\kappa \notin \mathcal{S}'$, the vertex v^ν_μ is connected to v'_ν is connected to v^κ is connected to v^κ_μ which is a contradiction to T being a tree. ☐

4 Trees that Minimize Distances

In this section, we consider the problem of computing spanning trees of given graphs that minimize distances among the vertices of the graph under certain matrix norms.

Problem:	DISTANCE MINIMIZING SPANNING TREE (with respect to $\|\cdot\|_r$)
Input:	A connected graph G and an algebraic number γ
Question:	Does G contain a spanning trees T with $\|D_T\|_r \leq \gamma$?

We also consider the *forced-edge* version of this problem. Here, the input is a graph G, an edge set $E_0 \subseteq E(G)$, and an algebraic number γ, and the question is whether there exists a spanning tree T such that $E_0 \subseteq E(T)$ and $\|D_T\|_r \leq \gamma$.

We begin with our study by proving that computing distance-minimizing spanning trees is computationally hard under the L_p matrix-norm for all reasonable $1 \leq p < \infty$. Note that the case $p = 1$ corresponds to the MAD-tree problem which was shown to be NP-complete in [17]. In fact, our proof generalizes the X3C-based proof technique from [17].

Theorem 5. DISTANCE MINIMIZING SPANNING TREE *with respect to* $\|\cdot\|_{L,p}$ *is* NP*-complete, for all* $p \in \mathbb{N}_+$.

Proof. Containment in NP is obvious. We prove the hardness by reduction from X3C using the graph representation $G_{a,b}(\mathcal{C}, L)$ for any X3C instance $(\mathcal{C}, L)$. We will fix the parameter a, b and γ in an appropriate manner later, so that $(\mathcal{C}, L)$ has an admissible solution $\mathcal{S} \subseteq \mathcal{C}$ if and only if $G_{a,b}(\mathcal{C}, L)$ has a spanning tree T such that $\|D_T\|^p_{L,p} \leq \gamma^p$.

Suppose $S \subseteq \mathcal{C}$ is an admissible solution to instance $(\mathcal{C}, L)$. Let $T_{\mathcal{S}}$ be the corresponding spanning tree of $G_{a,b}(\mathcal{C}, L)$. Define

$$N =_{\text{def}} \sum_{u \in R \cup C} \sum_{v \in V} d_{G_{a,b}(\mathcal{C},L)}(u,v)^p.$$

By Proposition 2, N remains unchanged if the distance in $G_{a,b}(\mathcal{C}, L)$ are replaced by the distances in $T_{\mathcal{S}}$. More precisely,

$$N = 2\left(\underbrace{2^p\frac{a^2-a}{2}}_{\text{R to R}} + \underbrace{a}_{\text{R to X}} + \underbrace{2^p sa}_{\text{R to C}} + \underbrace{3^p 3ma}_{\text{R to L}} + \underbrace{s}_{\text{X to C}} + \underbrace{2^p 3m}_{\text{X to L}}\right).$$

Define

$$M =_{\text{def}} \sum_{u,v \in C \cup L} d_{T_{\mathcal{S}}}(u,v).$$

We obtain

$$M = 2\left(\underbrace{2^p\frac{s^2-s}{2}}_{\text{C to C}} + \underbrace{3m + 3^p 3m(s-1)}_{\text{C to L}} + \underbrace{2^p 3m + 4^p\frac{9m^2-9m}{2}}_{\text{L to L}}\right).$$

We now set our parameter as follows:

$$\begin{aligned} a &=_{\text{def}} \left\lceil \frac{M}{4^p-3^p} \right\rceil \\ b &=_{\text{def}} 0 \\ \gamma &=_{\text{def}} (N+M)^{1/p} \end{aligned}$$

Clearly, $\|D_{T_{\mathcal{S}}}\|_{L,p}^p = N + M = \gamma^p$. Thus, $T_{\mathcal{S}}$ is a spanning tree of $G_{a,b}(\mathcal{C}, L)$ having the desired distance property.

Suppose T is a spanning tree of $G_{a,b}(\mathcal{C}, L)$ such that $\|D_T\|_{L,p}^p \leq \gamma^p$. We apply the characterization of a solution tree given in Lemma 3 and show that all conditions are satisfied. Note that, by Proposition 2, N is a lower bound for the p-distance sum between vertices in $R \cup X$.

- Assume to the contrary that the first condition of Lemma 3 does not hold, i.e., for some $\mu \in \{1, \ldots, s\}$, the edge $\{C_\mu, x\}$ is not in T. Then, $d_T(C_\mu, x) \geq 3$ and for all $\nu \in \{1, \ldots, 3m\}$, $d_T(C\mu, r_\nu) \geq 4$. Thus, $\|D_T\|_{L,p}^p \geq N - 1^p - 2^p a + 3^p + 4^p a$ and we conclude

$$\|D_T\|_{L,p}^p - \gamma^p \geq 3^p - 1 + a(4^p - 2^p) - M > 1 + M\frac{(4^p-2^p)}{4^p-3^p} - M > 1,$$

  a contradiction.

- Assume to the contrary that the second condition of Lemma 3 does not hold, i.e., there is a vertex l_μ not adjacent to any $\mathcal{C}_\nu$ in T. Then, $\|D_T\|_{L,p}^p \geq N - 2^p - 3^p a + 3^p + 4^p a$ and we conclude

$$\|D_T\|_{L,p}^p - \gamma^p \geq 3^p - 2^p + a(4^p - 3^p) - M > 1 + M\frac{(4^p-3^p)}{4^p-3^p} - M = 1,$$

  a contradiction.

- Note that, if the first and second condition in Lemma 3 are both satisfied, then all edges but those between $\mathcal{C}$ and L are already fixed by now and the distances in T and $G_{a,b}(\mathcal{C}, L)$ are the same except for those between vertices in L (between L and $\mathcal{C}$, each l_μ has p-distance one to exactly one C_ν and 3^p otherwise). Let g be the number of pairs (l_μ, l_ν) such that edges $\{l_\mu, C_\kappa\}$ and $\{l_\nu, C_\kappa\}$ exist in T. The total number of pairs is $9m^2 - 3m$. We obtain

  $$\sum_{\mu=1}^{3m}\sum_{\nu=1}^{3m} d_T(l_\mu, l_\nu)^p = 2^p g + 4^p(9m^2 - 3m - g).$$

  The maximum possible value of g is $6m$ which corresponds to the case that the third condition in Lemma 3 is satisfied. Assume to the contrary $g < 6m$. Then we have

  $$\begin{aligned}\|D_T\|_{L,p}^p - \gamma^p \geq -2^p 6m - 4^p(9m^2 - 9m) + 2^p g + 4^p(9m^2 - 3m - g) = \\ = (6m - g)(4^p - 2^p) > 1,\end{aligned}$$

  a contradiction.

This proves the theorem by applying Lemma 3. □

We know from the literature [8, 16] that a minimum diameter spanning tree in a graph can be found in time $O(mn + n^2 \log n)$ via computing absolute 1-centers. However, if we require that certain edges have to be in the spanning tree, feasibility becomes out of reach.

Theorem 6. *The forced-edge version of* DISTANCE MINIMIZING SPANNING TREE *with respect to* $\|\cdot\|_{L,\infty}$ *is* NP*-complete.*

Proof. Containment in NP is obvious. To show the NP-hardness, we give reduction from 2-HITTING SET based on the graph representation $G(\mathcal{C}, \mathcal{S}, k)$ for any given instance $(\mathcal{C}, \mathcal{S}, k)$ of 2-HITTING SET. Let N be the number of vertices of $G(\mathcal{C}, \mathcal{S}, k)$. Define G to be the graph constructed from $G(\mathcal{C}, \mathcal{S}, k)$ by adding

- vertices c, c', d, d',
- edges $\{c, c'\}$, $\{d, d'\}$, $\{c', a\}$, $\{b, d'\}$, and
- two paths P and Q connecting c, c' and d, d', respectively, each of length $N + 1$.

An illustration of G is given in Fig. 5. Define the set E' of edges that must be contained in any desired spanning tree to consist of all edges in clause paths of $G(\mathcal{C}, \mathcal{S}, k)$ and all edges in paths P and Q. That is, any spanning tree T of G with $E' \subseteq E(T)$ must include P and Q completely. Therefore, $d_T(c, d) \geq 2N$. For any vertices v, w in $G(\mathcal{C}, \mathcal{S}, k)$ and any vertex u which is G but not in $G(\mathcal{C}, \mathcal{S}, k)$, we have $d_T(v, w) \leq N - 1$ and $d_T(v, u) \leq 2N - 1$. All in all, this shows that the distance in T between vertices c and d determines the diameter of T, i.e., $\|D_T\|_{L,\infty} = d_T(c, d)$. Clearly, any path from vertex c to vertex d must pass vertices a and b. Consequently,

$$\|D_T\|_{L,\infty} = 2N + 2 + d_T(a, b).$$

Define $\gamma =_{\text{def}} 2N + 4 + n(m + 2) + k$ where $n = \|\mathcal{S}\|$ and $m = \|\mathcal{C}\|$ for a given 2-HITTING SET instance $(\mathcal{C}, \mathcal{S}, k)$. Using Lemma 4, we immediately see that $(\mathcal{C}, \mathcal{S}, k)$ has an admissible solution $\mathcal{S}' \subseteq \mathcal{S}$ if and only if there exists a spanning tree T in the graph G which includes all edges of E' and satisfies $\|D_T\|_{L,\infty} \leq \gamma$. This proves the theorem. □

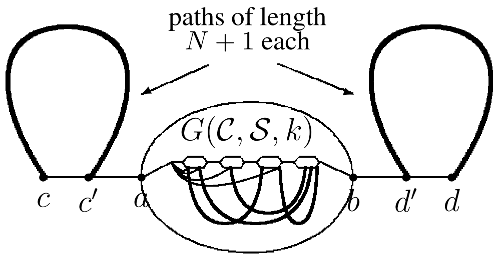

Figure 5: Construction for proving the NP-completeness of the fixed-edge version of DISTANCE MINIMIZING SPANNING TREE with respect to $\|\cdot\|_{L,\infty}$. The dashed bold paths consist solely of edges that are included in the edge set required to be in a spanning tree for G).

Next we state that distance-minimizing spanning trees are hard to find under the maximum column-sum norm and thus, maximum row-sum norm as well.

Theorem 7. DISTANCE MINIMIZING SPANNING TREE *with respect to* $\|\cdot\|_1$ *is* NP*-complete.*

Proof. Containment in NP is obvious. Again, NP-hardness is proven by reduction from X3C using the graph representation $G_{a,b}(\mathcal{C}, L)$ for a given X3C instance $(\mathcal{C}, L)$ and an appropriate choice of the parameters a, b and γ. We will fix the parameter later, so that $(\mathcal{C}, L)$ has an admissible solution $\mathcal{S} \subseteq \mathcal{C}$ if and only if $G_{a,b}(\mathcal{C}, L)$ has a spanning tree T such that $\|D_T\|_1\gamma$.

Suppose $\mathcal{S} \subseteq CF$ is an admissible solution to $(\mathcal{C}, L)$. Let $T_\mathcal{S}$ be the corresponding spanning tree in the graph $G_{a,b}(\mathcal{C}, L)$. The vertices in the sets R, X, and L all have the same column sums. We calculate for $\mu \in \{1, \ldots, a\}$ and $\nu \in \{1, \ldots, 3\}$:

$$\begin{aligned}
\sum_{v\in V} d_T(r_\mu, v) &= 2a + 2s + 9m - 1 \\
\sum_{v\in V} d_T(x, v) &= a + s + 6m \\
\sum_{v\in V} d_T(l_\nu, v) &= 3a + 3s + 12m - 8
\end{aligned}$$

For $\mathcal{C}$ we have to make a distinction between vertices with one neighbor in $T_\mathcal{S}$ or four:

$$\sum_{v\in V} d_T(C_\mu, v) = \begin{cases} 2a + 2s + 9m - 1 & \text{if } C_\mu \text{ has one neighbor in } T_\mathcal{S} \\ 2a + 2s + 9m - 7 & \text{if } C_\mu \text{ has four nighbors in } T_\mathcal{S} \end{cases}$$

We define our parameters as follows:

$$\begin{aligned}
a &=_{\text{def}} s + 12m + 8 \\
b &=_{\text{def}} 0 \\
\gamma &=_{\text{def}} 6s + 48m + 16
\end{aligned}$$

Clearly, we have $\|D_{T_\mathcal{S}}\|_1 = 6s + 48m + 16 = \gamma$. Thus, $T_\mathcal{S}$ is a spanning tree in $G_{a,b}(\mathcal{C}, L)$ having a distance property as desired.

Suppose T is a spanning tree in $G_{a,b}(\mathcal{C}, L)$ satisfying $\|D_T\|_1 \leq \gamma$. We apply the characterization of a solution tree given in Lemma 3 and show that all conditions are satisfied.

- Assume to the contrary that the first condition in Lemma 3 does not hold, i.e., for some $\mu \in \{1, \ldots, s\}$, the edge $\{C_\mu, x\}$ does not belong to T. We obtain

$$\sum_{v \in V} d_T(C_\mu, v) \geq 4a + 2s + 9m - 5 = 6s + 57m + 27 > \gamma,$$

a contradiction.

- Assume to the contrary that the second condition in Lemma 3 does not hold, i.e., there is a vertex l_μ not adjacent to any vertex C_ν in T. Then

$$\sum_{v \in V} d_T(l_\mu, v) \geq 4a + 2s + 3 = 6s + 48m + 35 > \gamma,$$

a contradiction.

- Assume to the contrary that the third condition in Lemma 3 does not hold, i.e., there is a vertex C_μ having two or three neighbors in T. Let l_ν be one of C_μ's neighbors in T. We calculate

$$\sum_{v \in V} d_T(l_\nu, v) \geq 3a + 3s + 12m - 6 = 6s + 48m + 18 > \gamma,$$

a contradiction.

This proves the theorem by Lemma 3. □

Remark 8. *Both constructions in Theorem 5 and Theorem 7 do not really use the edges between the vertices in L of the graph representation of an* X3C *instance. Consequently, the constructed graphs are planar (if we assume that all clauses in the* X3C *instance are distinct). That means, that computing distance-minimizing spanning trees for these norms is* NP*-hard already in planar graphs.*

5 Trees that Approximate Distances

In this section, we turn to the problem of finding spanning trees approximating the distances in a graph reasonably well under a certain given matrix norm.

Problem:	DISTANCE APPROXIMATING SPANNING TREE (with respect to $\|\cdot\|_r$)
Input:	A connected graph G and an algebraic number γ
Question:	Does G contain a spanning trees T with $\|D_T - D_G\|_r \leq \gamma$?

The forced-edge version of this problem is specified by the instance consisting of a graph G, edge set $E_0 \subseteq E(G)$, and an algebraic numbers γ and the question whether there exists a spanning tree T such that $E_0 \subseteq E(T)$ and $\|D_T - D_G\|_r \leq \gamma$. Clearly, the forced-edge version is algorithmically not easier than the original one.

Notice that with respect to L_1 matrix-norm, computing distance-minimizing and optimal distance-approximating spanning trees is equivalent. As a consequence, we immediately have NP-completeness under L_1 matrix-norm (from Theorem 5 or [17]).

We show that, independent of the norm, all problems are NP-complete.

Theorem 9. DISTANCE APPROXIMATING SPANNING TREE *with respect to* $\|\cdot\|_{L,p}$ *is* NP*-complete for all* $p \in \mathbb{N}_+$.

Proof. Containment in NP is obvious. NP-hardness is proven by reduction from X3C using the graph representation $G_{a,b}(\mathcal{C}, L)$ for a given X3C instance $(\mathcal{C}, L)$ and an appropriate choice of the parameters a, b and γ. We will fix the parameter later, so that $(\mathcal{C}, L)$ has an admissible solution $\mathcal{S} \subseteq \mathcal{C}$ if and only if $G_{a,b}(\mathcal{C}, L)$ has a spanning tree T such that $\|D_{G_{a,b}(\mathcal{C},L)} - D_T\|_{L,p}^p \leq \gamma^p$.

Suppose $\mathcal{S} \subseteq \mathcal{C}$ is an admissible solution to $(\mathcal{C}, L)$. Let $T_\mathcal{S}$ be the corresponding spanning tree in $G_{a,b}(\mathcal{C}, L)$. By Proposition 2, we only have $d_{T_\mathcal{S}}(l_\mu, l_\nu) - d_{G_{a,b}(\mathcal{C},L)}(l_\mu, l_\nu) > 0$ and $d_{T_\mathcal{S}}(l_\mu, C_\nu) - d_{G_{a,b}(\mathcal{C},L)}(l_\mu, C_\nu) > 0$. Thus,

$$\|D_{G_{a,b}(\mathcal{C},L)} - D_{T_\mathcal{S}}\|_{L,p}^p = 2\left(\underbrace{2^p 3m(s-m) + 3m(m-1)}_{\text{C to L}} + \underbrace{3m + 3^p \frac{9m^2 - 9m}{2}}_{\text{L to L}}\right).$$

We now set our parameters as follows:

$$\begin{aligned} a &=_{\text{def}} \gamma \\ b &=_{\text{def}} 0 \\ \gamma &=_{\text{def}} \|D_{G_{a,b}(\mathcal{C},L)} - D_{T_\mathcal{S}}\|_{L,p} \end{aligned}$$

Note that computing γ in polynomial time is possible, since all information needed is already given in the input. By definition, $T_\mathcal{S}$ is a spanning tree in $G_{a,b}(\mathcal{C}, L)$ having the desired distance property.

Suppose T is a spanning tree in $G_{a,b}(\mathcal{C}, L)$ satisfying $\|D_{G_{a,b}(\mathcal{C},L)} - D_T\|_{L,p}^p \leq \gamma^p$. We apply the characterization of a solution tree given in Lemma 3 and show that all conditions are satisfied.

- Assume to the contrary that the first condition in Lemma 3 does not hold, i.e., for some $\mu \in \{1, \ldots, s\}$, the edge $\{C_\mu, x\}$ does not belong to T. This implies $d_T(C_\mu, x) \geq d_{G_{a,b}(\mathcal{C},L)}(C_\mu, x) + 2$ and for all $\nu \in \{1, \ldots, a\}$, $d_T(C_\mu, r_\nu) \geq d_{G_{a,b}(\mathcal{C},L)}(C_\mu, r_\nu) + 2$. Thus,

  $$\|D_{G_{a,b}(\mathcal{C},L)} - D_T\|_{L,p}^p \geq (a+1)2^p > \gamma^p,$$

  a contradiction.

- Assume to the contrary that the second condition in Lemma 3 does not hold, i.e., there is a vertex l_μ not adjacent to any vertex C_ν in T. We obtain $d_T(l_\mu, x) \geq d_{G_{a,b}(\mathcal{C},L)}(l_\mu, x) + 1$ and for all $\nu \in \{1, \ldots, a\}$, $d_T(l_\mu, r_\nu) \geq d_{G_{a,b}(\mathcal{C},L)}(l_\mu, r_\nu) + 1$. This gives

  $$\|D_{G_{a,b}(\mathcal{C},L)} - D_T\|_{L,p}^p \geq a + 1 > \gamma^p,$$

  a contradiction.

- Note that, if the first and second condition in Lemma 3 are both satisfied, then all edges but those between $\mathcal{C}$ and L are already fixed by now and the distances in T and $G_{a,b}(\mathcal{C}, L)$ are the same. For the distances from vertices in $\mathcal{C}$ to vertices in L we have

  $$d_T(l_\mu, C_\nu) = \begin{cases} d_{G_{a,b}(\mathcal{C},L)}(l_\mu, C_\nu) & \text{if edge } \{l_\mu, c_\nu\} \text{ is in } T \\ d_{G_{a,b}(\mathcal{C},L)}(l_\mu, C_\nu) + 1 & \text{if edge } \{l_\mu, c_\nu\} \text{ is not in } T \text{ and } l_\mu \notin C_\nu \\ d_{G_{a,b}(\mathcal{C},L)}(l_\mu, C_\nu) + 2 & \text{if edge } \{l_\mu, c_\nu\} \text{ is not in } T \text{ and } l_\mu \in C_\nu \end{cases}$$

Let h_i be the number of vertices C_μ having exactly i neighbors in T. It clearly holds $h_1 + h_2 + h_3 + h_4 = s$ and $h_2 + 2h_3 + 3h_4 = 3m$. Moreover, we have

$$\sum_{\mu=1}^{3m}\sum_{\nu=1}^{s}\left(d_{G_{a,b}(\mathcal{C},L)}(l_\mu, C_\nu) - d_T(l_\mu, C_\nu)\right)^p = 2\left(3s(m-1) + 2^p(3h_1 + 2h_2 + h_3)\right).$$

For the distances between vertices in L, we obtain for $\mu, \nu \in \{1, \ldots, 3m\}$ and $\mu \neq \nu$,

$$d_T(l_\mu, l_\nu) = \begin{cases} d_{G_{a,b}(\mathcal{C},L)}(l_\mu, l_\nu) + 1 & \text{if edges } \{l_\mu, C_\kappa\} \text{ and } \{l_\nu, C_\kappa\} \text{ belong to } T \\ & \text{for some } \kappa \in \{1, \ldots, s\} \\ d_{G_{a,b}(\mathcal{C},L)}(l_\mu, l_\nu) + 3 & \text{otherwise} \end{cases}$$

Let g be the number of pairs (l_μ, l_ν) such that $\mu \neq \nu$ and for some $\kappa \in \{1, \ldots, s\}$, edges $\{l_\mu, C_\kappa\}$ and $\{l_\mu, C_\kappa\}$ exist in T. We calculate

$$\sum_{\mu=1}^{3m}\sum_{\nu=1}^{3m}\left(d_{G_{a,b}(\mathcal{C},L)}(l_\mu, l_\nu) - d_T(l_\mu, l_\nu)\right)^p = g + 3^p(9m^2 - 3m - g).$$

The maximum possible value of g is $6m$ and that of h_4 is s both values simultaneously corresponding to the case that the third condition in Lemma 3 is satisfied. Assume to the contrary $g < 6m$ and $h_4 < s$. Then we have

$$\begin{aligned} &\|D_{G_{a,b}(\mathcal{C},L)} - D_T\|_{L,p}^p - \gamma^p \geq \\ &\quad - 6s(m-1) - 6m - 3^p(9m^2 - 9m) + 6s(m-1) + 2^{p+1} + g + 3^p(9m^2 - 3m - g) \\ &\qquad\qquad = 2^{p+1} + (3^p - 1)(6m - g) > 1, \end{aligned}$$

a contradiction.

This proves the theorem by Lemma 3. □

For proving NP-completeness for L_∞ matrix-norm, it is helpful to prove a completeness result first for the forced-edge version.

Lemma 10. *The forced-edge version of* DISTANCE APPROXIMATING SPANNING TREE *with respect to* $\|\cdot\|_{L,\infty}$ *is* NP*-complete.*

Proof. The proof is the same as the one for DISTANCE MINIMIZING SPANNING TREE with respect to $\|\cdot\|_{L,\infty}$ (see Theorem 6). The only difference in the reduction from 2-HITTING SET is that the parameter γ is now defined as $2N + k$ (what is clear to follow from Lemma 4) for a 2-HITTING SET instance $(\mathcal{C}, \mathcal{S}, k)$. □

We now try to get rid of the forced edges. In order to achieve this, we replace forced edges by cycles such that deleting a forced edge will cause the distance between two cycle vertices to increase by more than the allowed threshold γ. A similar technique with two cycles was used in [6, 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.

Lemma 11. *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 $\|D_T - D_G\|_{L,\infty} \le k$ if and only if there exists a spanning tree T' of G' such that $\|D_{T'} - D_{G'}\|_{L,\infty} \le k$.*

Proof. For any spanning tree T of a graph $G = (V, E)$, we define $\delta_T(v) =_{\text{def}} \max_{w \in V}(d_T(v, w) - d_G(v, w))$. Let P be the path $(v, u_1, \ldots, u_k, w)$ to be added to the graph $G = (V, E)$ with respect to the edge $\{v, w\}$. That is, $G' = G \cup P$. We prove the two directions separately.

($\Rightarrow$) Suppose there is a spanning tree T of G such that $\|D_T - D_G\|_{L,\infty} \le k$ and edge $\{v, w\}$ belongs to T. Without loss of generality, we assume that $\delta_T(v) \le \delta_T(w)$. Define T' to be the spanning tree in G' with edge set $E(T) \cup E(P)$ by removing the edge $\{u_{\lfloor \frac{k}{2} \rfloor}, u_{\lceil \frac{k+1}{2} \rceil}\}$ in the middle of P. We have two cases.

- Suppose $\delta_T(v) < k$. We have the following bounds on distance changes in T' with respect to G'.
  - For $x, y \in V(G)$ we have $d_{T'}(x, y) \le d_{G'}(x, y) + k$.
  - For $x, y \in V(P)$ we have $d_{T'}(x, y) \le d_{G'}(x, y) + k$.
  - For $\mu \in \{1, \ldots, \lfloor \frac{k}{2} \rfloor\}$ and $y \in V(G)$ we find

$$\begin{aligned} d_{T'}(u_\mu, y) - d_{G'}(u_\mu, y) &= d_{T'}(u_\mu, v) + d_{T'}(v, y) - d_{G'}(u_\mu, v) - d_{G'}(v, y) \\ &= d_{T'}(v, y) - d_{G'}(v, y) \;\; \le \;\; k. \end{aligned}$$

  - For $\mu \in \{\lceil \frac{k+1}{2} \rceil, \ldots, k\}$ and $y \in V(G)$, we have a similar inequality, if the shortest path from u_μ to y in G' contains vertex w. Otherwise, we obtain

$$\begin{aligned} d_{T'}(u_\mu, y) - d_{G'}(u_\mu, y) &= d_{T'}(u_\mu, v) + d_{T'}(v, y) - d_{G'}(u_\mu, v) - d_{G'}(v, y) \\ &= 1 + d_{T'}(v, y) - d_{G'}(v, y) \;\; \le \;\; 1 + (k - 1) \;\; = \;\; k. \end{aligned}$$

  This completes the first case.

- Suppose $\delta_T(v) = \delta_T(w) = k$. For $k \ge 0$ and for any vertex $z \in V$, define $B_{=k}(z) =_{\text{def}} \{x \in V \mid d_T(x, z) - d_G(x, z) = k\}$. First, we consider vertices $x, y \in B_{=k}(v) \cup B_{=k}(w)$ and claim that
  - either $d_T(v, x) = d_T(w, x) + 1$ and $d_T(v, y) = d_T(w, y) + 1$
  - or $d_T(w, x) = d_T(v, x) + 1$ and $d_T(w, y) = d_T(v, y) + 1$.

  Assume to the contrary that this is not true, i.e., we have $d_T(v, x) = d_T(w, x) + 1$ and $d_T(w, y) = d_T(v, y) + 1$. (By symmetry, it is enough consider this situation.) Now we may conclude that a path from x to y in T must pass v and w where x is nearer to w and y is nearer to v. Hence,

$$\begin{aligned} & d_T(x, y) - d_G(x, y) \\ &= d_T(x, w) + 1 + d_T(v, y) - d_G(x, y) \\ &\ge d_T(x, w) + 1 + d_T(v, y) - d_G(x, w) - d_G(v, y) - 1 && \text{(by triangle inequality)} \\ &\ge d_T(x, w) - d_G(x, w) + d_T(v, y) - d_G(v, y) - 2 && \text{(edge } \{v, w\} \text{ belongs to } T\text{)} \\ &\ge 2k - 4 && \text{(since } x, y \in B_{=k}(v) \cup B_{=k}(w)\text{)} \end{aligned}$$

For $k > 4$ this leads to a contradiction and thus, our claim is true in this case. The case $k \leq 4$ will be treated separately below.

So, without loss of generality, we suppose that $d_T(v, x) = d_T(w, x) + 1$ and $d_T(v, y) = d_T(w, y) + 1$. We obtain the following distance changes in T' with respect to G'

– For $x, y \in V(G)$ we trivially have $d_{T'}(u_\mu, y) \leq d_{G'}(u_\mu, y) + k$.
– For $\mu \in \{1, \ldots, \lfloor \frac{k}{2} \rfloor\}$ and $y \in V(G)$, the shortest path between u_μ and y visits v. Thus, $d_{T'}(u_\mu, y) \leq d_{G'}(u_\mu, y) + k$.
– For $\mu \in \{\lceil \frac{k+1}{2} \rceil + 1, \ldots, k\}$ and $y \in V(G)$, the shortest path between u_μ and y visits w. Thus, $d_{T'}(u_\mu, y) \leq d_{G'}(u_\mu, y) + k$.
– For $\mu = \lceil \frac{k+1}{2} \rceil$ and $y \in B_{=k}(v) \cup B_{=k}(w)$ we know from above that the shortest path between u_μ and y visits w and hence, $d_{T'}(u_\mu, y) \leq d_{G'}(u_\mu, y) + k$. For $y \notin B_{=k}(v) \cup B_{=k}(w)$ we obtain

$$\begin{array}{rcl} d_{T'}(u_\mu, y) - dG'(u_\mu, y) & = & d_{T'}(u_\mu, v) + d_{T'}(v, y) - d_{G'}(u_\mu, v) - d_{G'}(v, y) \\ & \leq & 1 + (k-1) \quad = \quad k \end{array}$$

Finally, for $k = 4$, note that $d_T(u_\mu, v) = d_G(u_\mu, v)$ and $d_T(u_\mu, w) = d_G(u_\mu, w)$, if we remove the edge $\{u_2, u_3\}$. Hence,

$$d_{T'}(u_\mu, y) - d_{G'}(u_\mu, y) = \min\left(d_T(v, y) - d_G(v, y), d_T(w, y) - d_G(w, y)\right) \leq k.$$

This completes the second case.

($\Leftarrow$) Suppose there is a spanning tree T' for G' with $\|D_{T'} - D_{G'}\|_{L,\infty} \leq k$. We show that for any such tree, the edge $\{v, w\}$ must be in T'. Note that $\{v, w\}$ is in at least two cycles in G', where one is a cycle with P and another one is the cycle making $\{v, w\}$ a non-bridge-edge in G. These cycles must be broken in order for T' to be a tree. We show that there is only one possibility to break these cycles (see Fig. 6 for illustration):

- Breaking the cycle in P at $\{u_\mu, u_{\mu+1}\}$ and the cycles in G at $\{v, w\}$ yields

$$\begin{array}{rcl} d_{T'}(u_\mu, u_{\mu+1}) - d_{G'}(u_\mu, u_{\mu+1}) & = & d_{T'}(u_\mu, u_{\mu+1}) - 1 \\ & = & d_{T'}(u_\mu, v) + d_{T'}(v, w) + d_{T'}(w, u_{\mu+1}) - 1 \\ & \geq & d_{T'}(u_\mu, v) + 2 + d_{T'}(w, u_{\mu+1}) - 1 \\ & = & k + 1 \quad > \quad k, \end{array}$$

  a contradiction.

- Breaking the cycles in P at $\{v, w\}$ and any of the other one at an arbitrary edge, say $\{x, y\}$ with $y \notin \{v, w\}$ yields

$$\begin{array}{rcl} d_{T'}(x, y) - d_{G'}(x, y) & = & d_{T'}(x, y) - 1 \\ & = & d_{T'}(x, v) + d_{T'}(v, w) + d_{T'}(w, y) - 1 \\ & \geq & d_{T'}(x, v) + k + 1 \quad > \quad k, \end{array}$$

  again a contradiction.

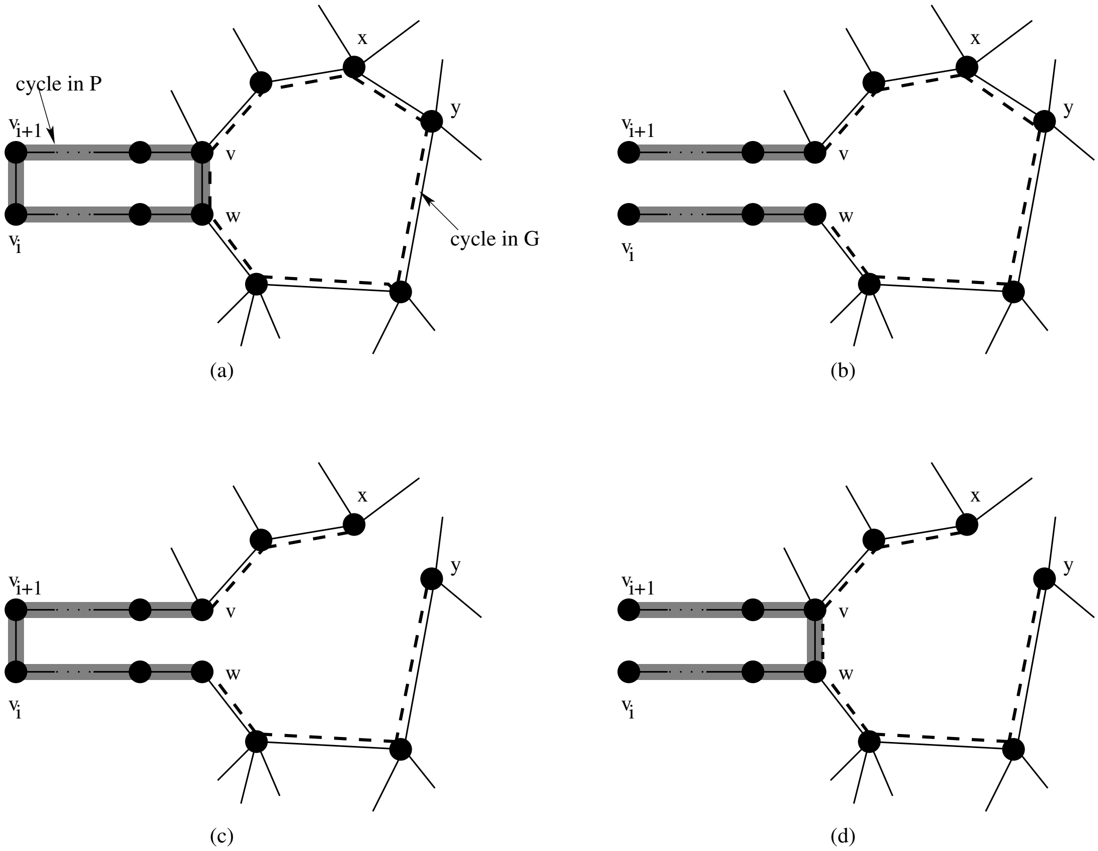

Figure 6: Illustration of the proof of Lemma 11. Only T_3 has distance difference $\leq k$ for $i = \lfloor \frac{k}{2} \rfloor$.

It follows that when breaking the cycle in G at any edge $e \neq \{u, w\}$ and the cycle with P at the edge $\{u_{\lfloor \frac{k}{2} \rfloor}, u_{\lceil \frac{k+1}{2} \rceil}\}$, we can "reverse" the assembly - that is we can omit the part of T' that spans P, and thus obtain a tree T of G for which $\|D_T - D_G\|_{L,\infty} \leq k$ and $\{v, w\}$ is an edge in T. $\square$

From these two lemmas we easily obtain our result concerning the L_∞ matrix-norm.

Theorem 12. DISTANCE APPROXIMATING SPANNING TREE *with respect to* $\|\cdot\|_{L,\infty}$ *is* NP*-complete*

Proof. Using Lemma 10, we prove the NP-hardness by a reduction from the fixed-edge version of DISTANCE APPROXIMATING SPANNING TREE. We may restrict ourselves to instance (G, γ, E') with $\gamma > 3$. First, note that if E' contains any bridges, we may remove these from E' without changing the optimum solution to the given instance, as a bridge must be contained in any spanning tree of G. Second, if some edges in E' form a cycle, we may immediately reject the instance. Using Lemma 11 we describe the following reduction: for every edge $\{v, w\} \in E'$ iteratively add a path $(v, u_1, \ldots, u_k, w)$ with new vertices. Let G' be the resulting graph which of course can constructed in polynomial time in the size of G. An easy induction on the size of E' now shows that G has a spanning tree containing all edges of E' such that $\|D_T - D_G\|_{L,\infty} \leq \gamma$ if and only if G' has a spanning tree T' such that $\|D_{T'} - D_{G'}\|_{L,\infty} \leq \gamma$. $\square$

We finally show the hardness with respect to the maximum column-sum norm.

Theorem 13. DISTANCE APPROXIMATING SPANNING TREE *with respect to* $\|\cdot\|_1$ *is* NP*-complete.*

Proof. Containment in NP is obvious. We prove NP-hardness by reduction from X3C using the graph representation $G_{a,b}(\mathcal{C},L)$ for a given X3C instance $(\mathcal{C},L)$ and an appropriate choice of the parameters a,b and γ. We will fix the parameter later, so that $(\mathcal{C},L)$ has an admissible solution $\mathcal{S}\subseteq\mathcal{C}$ if and only if $G_{a,b}(\mathcal{C},L)$ has a spanning tree T such that $\|D_{G_{a,b}(\mathcal{C},L)}-D_T\|_1\leq\gamma$. We may suppose that $s\geq 1$ and thus $m\geq 1$. For each $\mu\in\{1,\dots,3m\}$, let $h(\mu)$ denote the number of sets of $\mathcal{C}$ in which l_μ appears, i.e., $h(\mu)=\|\{\nu\mid l_\mu\in C_\nu\}\|$. Define $h_{max}=_{\text{def}}\max_\mu h(\mu)$.

Suppose $\mathcal{S}\subseteq\mathcal{C}$ is an admissible solution to $(\mathcal{C},L)$. Let $T_\mathcal{S}$ be the corresponding spanning tree in $G_{a,b}(\mathcal{C},L)$. The vertices in the sets R, X, L, and K all have the same column sums. We calculate for $\mu\in\{1,\dots,s\}$, $\nu\in\{1,\dots,3m\}$, and $\kappa\in\{1,\dots,b\}$:

$$\begin{aligned}
\sum_{v\in V} d_T(r_\mu,v)-d_{G_{a,b}(\mathcal{C},L)}(r_\mu,v) &= 0\\
\sum_{v\in V} d_T(x,v)-d_{G_{a,b}(\mathcal{C},L)}(x,v) &= 0\\
\sum_{v\in V} d_T(l_\nu,v)-d_{G_{a,b}(\mathcal{C},L)}(l_\nu,v) &= s+h(\nu)+9m-9+b(9m-7)\\
\sum_{v\in V} d_T(k_{\nu,\kappa},v)-d_{G_{a,b}(\mathcal{C},L)}(k_{\nu,\kappa},v) &= s+h(\nu)+9m-9+b(9m-7)
\end{aligned}$$

For vertices in $\mathcal{C}$ we have to make a distinction between vertices with one neighbor in T and vertices with four neighbors in T. We obtain for $\mu\in\{1,\dots,s\}$:

$$\sum_{v\in V} d_T(C_\mu,v)-d_{G_{a,b}(\mathcal{C},L)}(C_\mu,v)=\begin{cases}(3m+3)(b+1) & \text{if } C_\mu \text{ has one neighbor in } T_\mathcal{S}\\ (3m-3)(b+1) & \text{if } C_\mu \text{ has four neighbors in } T_\mathcal{S}\end{cases}$$

We set our parameters in the following way:

$$\begin{aligned}
a &=_{\text{def}} \gamma+1\\
b &=_{\text{def}} s+1\\
\gamma &=_{\text{def}} s+h_{\max}+9m-9+b(9m-7)
\end{aligned}$$

This gives $\|D_{G_{a,b}(\mathcal{C},L)}-D_T\|_1=s+h_{\max}+9m-9+b(9m-7)=\gamma$. Consequently, $T_\mathcal{S}$ is a spanning tree of $G_{a,b}(\mathcal{C},L)$ having the desired distance property.

Suppose T is a spanning tree in $G_{a,b}(\mathcal{C},L)$ satisfying $\|D_{G_{a,b}(\mathcal{C},L)}-D_T\|_1\leq\gamma$. We apply the characterization of a solution tree given in Lemma 3 and show that all conditions are satisfied.

- Assume to the contrary that the first condition in Lemma 3 does not hold, i.e., for some $\mu\in\{1,\dots,s\}$, the edge $\{C_\mu,x\}$ does not belong to T. Then

  $$\sum_{v\in V} d_T(C_\mu,v)-d_{G_{a,b}(\mathcal{C},L)}(C_\mu,v)\geq 4a>\gamma,$$

  a contradiction.

- Assume to the contrary that the second condition in Lemma 3 does not hold, i.e., there is a vertex l_μ not adjacent to any vertex C_ν in T. Then

$$\sum_{v \in V} d_T(l_\mu, v) - d_{G_{a,b}(\mathcal{C},L)}(l_\mu, v) \geq a > \gamma,$$

  a contradiction.

- Note that, if the first and second condition in Lemma 3 are both satisfied, then all edges but those between $\mathcal{C}$ and L are already fixed by now and the distances in T and $G_{a,b}(\mathcal{C}, L)$ are the same. Assume to the contrary that the third condition in Lemma 3 does not hold, i.e., there is a vertex C_μ having two or three neighbors in T. Let l_ν be a neighbor of such a vertex C_μ and let $\deg_T(C_\mu)$ denote the number of neighbors of C_μ in T. Then, we conclude

$$\begin{aligned}
&\sum_{v \in V} d_T(l_\nu, v) - d_{G_{a,b}(\mathcal{C},L)}(l_\nu, v) \\
&\quad = \quad s + h(l_\nu) - 2 + \deg_T(C_\mu) - 2 + 3(3m - \deg_T(C_\mu) + 1)(b+1) \\
&\quad = \quad \gamma - s - h_{max} - 9m + 9 - b(9m-7)s + h(l_\nu) - 4 + \deg_T(C_\mu) + \\
&\qquad\qquad + 3(3m - \deg_T(C_\mu) + 1)(b+1) \\
&\quad = \quad \gamma + (h(l_\nu) - h_{max}) + 8 - 2\deg_T(C_\mu) + b(10 - 3\deg_T(C_\mu)) \\
&\quad \geq \quad \gamma - s + b \ > \ \gamma,
\end{aligned}$$

  a contradiction.

This proves the theorem by Lemma 3. □

6 Trees that Approximate Centralities

A centrality measure is a mapping from the vertices of a graph to real numbers. Closeness centrality $c_G : V \to \mathbb{R}$ for a graph G is defined for all $v \in V$ as follows [2, 26]:

$$c_G(v) =_{\text{def}} \left(\sum_{t \in V} d_G(v, t) \right)^{-1}$$

It is clear from the definition that for each subgraph G' of a graph G, we have $c_G(v) \geq c_{G'}(v)$ for all vertices v in the graph.

We are interested in the problem of computing a spanning tree of a given graph such that its centrality function is as close as possible to the centrality function of the graph under some vector norms. We consider the following decision version of that problem.

Problem:	CLOSENESS APPROXIMATING SPANNING TREE (with respect to $\|\cdot\|_r$)
Input:	A graph G (not necessarily connected) and an algebraic number γ
Question:	Does G contain a spanning tree T with $\|c_G - c_T\|_r \leq \gamma$?

It turns out that computing trees approximating the closeness centrality best possible with respect to the average deviation is computationally hard.

Theorem 14. CLOSENESS APPROXIMATING SPANNING TREE *with respect to* $\|\cdot\|_1$ *is* NP*-complete.*

Proof. Containment in NP is obvious. We prove NP-hardness by reduction from X3C using a graph representation slightly different to the one we used so far. The difference lies in the following: the graph representation $G_{a,b}(\mathcal{C}, L)$ for an X3C instance $(\mathcal{C}, L)$ has edges $\{l_\mu, l_\nu\}$ for all pairs of literal vertices. In our new graph representation $G^*_{a,b}(\mathcal{C}, L) = (V^*, E^*)$ we omit these edges, i.e., we have

$$\begin{aligned} V^* &= V \\ E^* &= E \setminus \big\{\{l_\mu, l_\nu\} \mid \mu, \nu \in \{1, \ldots, 3m\} \text{ and } \mu \neq \nu\big\} \end{aligned}$$

where $G_{a,b}(\mathcal{C}, L) = (V, E)$. It is easy to see that Lemma 3 also holds for the new graph representation. Later we will set the parameters a, b and γ in a way that $(\mathcal{C}, L)$ has an admissible solution $\mathcal{S} \subseteq \mathcal{C}$ if and only if $G^*_{a,b}(\mathcal{C}, L)$ has a spanning tree T such that $\|c_{G^*_{a,b}(\mathcal{C},L)} - c_T\|_1 \leq \gamma$. In the following we may restrict ourselves to the cases where $m \geq 5$.

Suppose $\mathcal{S} \subseteq \mathcal{C}$ is an admissible solution to $(\mathcal{C}, L)$. Let $T_\mathcal{S}$ be the corresponding spanning tree in $G^*_{a,b}(\mathcal{C}, L)$. We obtain

$$c_{T_\mathcal{S}}(v)^{-1} = \begin{cases} 2s + 3(3+4b)m + 2a - 1 & \text{if } v \in R \\ s + 3(2+3b)m + a & \text{if } v \in X \\ 2s + 3(3+4b)m + 2a - 6(b+1) - 1 & \text{if } v \in \mathcal{S} \\ 2s + 3(3+4b)m + 2a - 1 & \text{if } v \in \mathcal{C} \setminus \mathcal{S} \\ 3s + 3(4+5b)m + 3a - 8(b+1) & \text{if } v \in L \\ 4s + 3(5+6b)m + 4a - 8(b+1) - 1 & \text{if } v \in K \end{cases}$$

We set our parameters as follows:

$$\begin{aligned} a &=_{\text{def}} 3s(b+1) + 3m(s-1)(b+1) + 3m(m-1)(b+1)^2 \\ b &=_{\text{def}} 9s + 1 \\ \gamma &=_{\text{def}} \|c_{G^*_{a,b}(\mathcal{C},L)} - c_{T_\mathcal{S}}\|_1 \end{aligned}$$

Thus, $T_\mathcal{S}$ is a spanning tree of $G^*_{a,b}(\mathcal{C}, L)$ having the desired centrality property. Note that all parameters and the graph representation $G^*_{a,b}(\mathcal{C}, L)$ can be computed in polynomial time in the size of $(\mathcal{C}, L)$. In particular, it is not necessary to know exactly the vertices of $\mathcal{S}$.

Suppose that T is a spanning tree of $G^*_{a,b}(\mathcal{C}, L)$ satisfying $\|c_{G^*_{a,b}(\mathcal{C},L)} - c_T\|_1 \leq \gamma$. We compare the centrality of each vertex in the tree T with its centrality in a hypothetical solution tree for the X3C instance $(\mathcal{C}, L)$. For $v \in V$, define imitating centralities $\hat{c}(v)$ as follows: if $v \in V \setminus \mathcal{C}$, then $\hat{c}(v)$ is equal to the values $c_{T_\mathcal{S}}$ from above; for vertices $v \in \mathcal{C}$, we define

$$\hat{c}(v)^{-1} =_{\text{def}} \begin{cases} 2s + 3(3+4b)m + 2a - 6(b+1) - 1 & \text{if } v \in \{C_1, \ldots, C_m\} \\ 2s + 3(3+4b)m + 2a - 1 & \text{if } v \in \{C_{m+1}, \ldots, C_s\}, \end{cases}$$

i.e., the clause vertices $C_1, \ldots, C_m$ simulate an admissible solution to $(\mathcal{C}, L)$. Note that $\|c_{G^*_{a,b}(\mathcal{C},L)} - \hat{c}\|_1 = \gamma$. We apply the characterization of a solution tree in Lemma 3 (in the version suitable for the graph representation $G^*_{a,b}(\mathcal{C}, L)$) and show that all conditions are satisfied.

- Assume to the contrary that the first condition in Lemma 3 does not hold, i.e., for some $\mu \in \{1, \ldots, s\}$, the edge $\{C_\mu, x\}$ does not belong to T. Simple calculations yield the following bounds on deviations from the imitating centralities.

- For $v \in R \cup X$ we obtain $c_T(v)^{-1} \geq \hat{c}(v)^{-1} + 2$. Note that this inequality is crucial in getting a contradiction as it holds for $a + 1$ vertices.
- For $v \in \mathcal{C}$ we obtain $c_T(v)^{-1} \geq \hat{c}(v)^{-1} - 6(b + 1)$.
- For $v \in L \cup K$, we have $c_T(v)^{-1} \geq \hat{c}(v)^{-1} - 2(s - 1) - 6(m - 1)(b + 1)$.

Thus, using the identity $\frac{1}{x+y} = \frac{1}{x} - \frac{y}{x(x+y)}$ which is at least true whenever $x > 0$ and $y \neq -x$, the total centrality of T can be estimated as

$$\sum_{v \in V} c_T(v) \leq \left(\sum_{v \in V} \hat{c}(v) \right) - \frac{2(a+1)}{(a + c_1)(a + c_2)} + \frac{A}{(2a + c_3)(2a + c_4)}$$

where c_1, c_2, c_3, c_4 and A are appropriate positive integers (that depend on s, m, and b). It is clear that the latter sum in the inequality is negative for a large enough. Inspecting the concrete values

$$\begin{aligned}
c_1 &= s + 3(2 + 3b)m \\
c_2 &= s + 3(2 + 3b)m + 2 \\
c_3 &= 2s + 3(3 + 4b)m - 6(b + 1) - 1 \\
c_4 &= 2s + 3(3 + 4b)m - 12(b + 1) - 1 \\
A &= 6s(b + 1) + 6m(s - 1)(b + 1) + 18m(m - 1)(b + 1)^2,
\end{aligned}$$

we see that $0 \leq c_1 \leq c_3$ and $0 \leq c_2 \leq c_4$ for $m \geq 5$. Thus, our choice of a from above is appropriate. Hence,

$$\|c_{G^*_{a,b}(\mathcal{C},L)} - c_T\|_1 > \|c_{G^*_{a,b}(\mathcal{C},L)} - \hat{c}\|_1 = \gamma,$$

a contradiction.

- The second condition of Lemma 3 holds because T is a spanning tree of $G^*_{a,b}(\mathcal{C}, L)$.

- Note that, if the first and second condition in Lemma 3 are both satisfied, then all edges but those between $\mathcal{C}$ and L are already fixed by now and the distances in T and $G_{a,b}(\mathcal{C}, L)$ are the same. Assume to the contrary that the third condition in Lemma 3 does not hold, i.e., there is a vertex C_μ having two or three neighbors in T. Let $\deg_T(v)$ denote the degree of vertex v in T. We consider several cases:

  - For $v \in R \cup X$ we clearly obtain $c_T(v)^{-1} = \hat{c}(v)^{-1}$.
  - For $v \in \mathcal{C}$ we have $c_T(v)^{-1} \geq \hat{c}(v)^{-1} - 6(b + 1)$.
  - For $v \in L$ it suffices to have $c_T(v)^{-1} \geq \hat{c}(v)^{-1}$.
  - For $v \in K$ we obtain $c_T(v)^{-1} \geq \hat{c}(v)^{-1} + 2(b + 1)(4 - \deg_T(u))$ where $u \in \mathcal{C}$ and T contains edges $\{v, w\}$ and $\{w, u\}$ for some $w \in L$. Note that since there is a vertex in $\mathcal{C}$ with at most three neighbors in T, there are at least b vertices in K such that $c_T(v)^{-1} \geq \hat{c}(v)^{-1} + 2(b + 1)$. This is the crucial inequality in getting a contradiction.

Using the identity $\frac{1}{x+y} = \frac{1}{x} - \frac{y}{x(x+y)}$ from above once more, we get the following estimation for the total centrality:

$$\sum_{v\in V} c_T(v) \leq \left(\sum_{v\in V} \hat{c}(v)\right) + \frac{6s(b+1)}{\hat{c}(v_0)^{-1}(\hat{c}(v_0)^{-1} - 6(b+1))} - \\ - \frac{2(b+1)b}{\hat{c}(u_0)^{-1}(\hat{c}(u_0)^{-1} + 4(b+1))},$$

where $v_0 \in \mathcal{C}$ and $u_0 \in K$. An easy estimation of the relation between $\hat{c}(v_0)^{-1}$ and $\hat{c}(u_0)^{-1}$ shows that, for $m \geq 5$, the latter difference is at most

$$\frac{18s(b+1) - 2(b+1)b}{\hat{c}(u_0)^{-1}(\hat{c}(u_0)^{-1} + 4(b+1))} < 0,$$

by our choice of b. Hence,

$$\|c_{G^*_{a,b}(\mathcal{C},L)} - c_T\|_1 > \|c_{G^*_{a,b}(\mathcal{C},L)} - \hat{c}\|_1 = \gamma,$$

a contradiction.

This proves the theorem by Lemma 3. □

Corollary 15. CLOSENESS APPROXIMATING SPANNING TREE *with respect to* $\|\cdot\|_{L,1}$ *is* NP*-complete, even restricted to planar graphs.*

Proof. Observe that the graph representation used in the proof of Theorem 14 always produces planar graphs, if X3C instance are assumed not to contain two or more identical clauses. □

7 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 (distance minimization, distance approximation, and centrality approximation). This provides the first computational complexity study in this area, presented in a unifying framework. Notably, all problems are NP-complete independent of the norms used, except for the case of minimizing distances with respect to the L_∞ matrix-norm which is the polynomial-time solvable minimum diameter spanning tree problem [8, 16].

Although we do not exactly understand at this point what makes the problems computationally hard, a simple observation is that the more closely the sub-trees and the graphs are related the more likely the problem is hard. As an example, it is harder to approximate the distances of the original graph (DISTANCE APPROXIMATING SPANNING TREE) than to optimize the tree itself (DISTANCE MINIMIZING SPANNING TREE) with respect to the L_∞ norm. On the other hand, as we have shown, DISTANCE MINIMIZING SPANNING TREE becomes NP-complete if some edges of the original graph have to be kept. For instance, ILP formulations of DISTANCE MINIMIZING SPANNING TREE and DISTANCE APPROXIMATING SPANNING TREE with respect to $\|\cdot\|_{L,\infty}$ are not very different, however, the first problem leads to an ILP with a homogenous set of constraint inequalities whereas the latter problem's set of constraint inequalities is inhomogenous. This not only provides a promising way for a better understanding of differences of these problems but also to approximation algorithms. A direct link between the subgraph and the given graph is essential for network abstraction (independent of the pattern classes and similarity measures) and it seems that it is exactly this link that also makes the problems hard. Hence, it is inevitable to look for approximation algorithms, exponential algorithms with small bases, or fixed-parameter algorithms.

An interesting technical problem is also left open. No results are known with respect to the spectral norm $\|\cdot\|_2$, i.e., $\|A\|_2 = \lambda_{\max}(A)$ where $\lambda_{\max}(A)$ is the largest eigenvalue of any symmetric matrix $A \in \mathbb{R}^{n\times n}$. Notice that for any symmetric matrix A, we have $\|A\|_2 \leq \|A\|_{L,\infty} \leq \|A\|_{L,p}$ for all $p \in \mathbb{N}_+$ and $\|A\|_2 \leq \|A\|_{L,\infty} \leq \|A\|_1 = \|A\|_\infty$. Can we expect that computing distance-minimizing spanning trees with respect to $\|\cdot\|_2$ is polynomial-time solvable (in the light that NP-completeness appears with coarser norms) ?

Acknowledgments. We thank Klaus Holzapfel and Alexander Offtermatt-Souza for helpful discussions and hints.

References

[1] Baruch Awerbuch. Complexity of network synchronization. *Journal of the ACM*, 32(4):804–823, 1985.

[2] Murray A. Beauchamp. An improved index of centrality. *Behavioral Science*, 10:161–163, 1965.

[3] Ulrik Brandes and Thomas Erlebach, editors. *Network Analysis: Methodological Foundations*, volume 3418 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[4] Ulrik Brandes and Dagmar Handke. NP-completeness results for minimum planar spanners. *Discrete Mathematics & Theoretical Computer Science*, 3(1):1–10, 1998.

[5] Andreas Brandstädt, Victor Chepoi, and Feodor Dragan. Distance approximating trees for chordal and dually chordal graphs. *Journal of Algorithms*, 30(1):166–184, 1999.

[6] Leizhen Cai. NP-completeness of minimum spanner problems. *Discrete Applied Mathematics*, 48(2):187–194, 1994.

[7] Leizhen Cai and Derek G. Corneil. Tree spanners. *SIAM Journal on Discrete Mathematics*, 8(3):359–387, 1995.

[8] Paolo M. Camerini, Giulia Galbiati, and Francesco Maffioli. Complexity of spanning tree problems: Part I. *European Journal of Operational Research*, 5(5):346–352, 1980.

[9] L. Paul Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, 1989.

[10] Elias Dahlhaus, Peter Dankelmann, Wayne Goddard, and Henda C. Swart. MAD trees and distance-hereditary graphs. *Discrete Applied Mathematics*, 131(1):151–167, 2003.

[11] Michael Elkin and David Peleg. $(1+\epsilon,\beta)$-spanner constructions for general graphs. *SIAM Journal on Computing*, 33(3):608–631, 2004.

[12] David Eppstein. Spanning trees and spanners. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. Elsevier, 2000.

[13] Sándor P. Fekete and Jana Kremer. Tree spanners in planar graphs. *Discrete Applied Mathematics*, 108(1–2):85–103, 2001.

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

[15] Dagmar Handke. *Graphs with Distance Guarantees*. Doctoral dissertation, Universität Konstanz, Fakultät für Mathematik und Informatik, December 1999.

[16] Refael Hassin and Arie Tamir. On the minimum diameter spanning tree problem. *Information Processing Letters*, 53(2):109–111, 1995.

[17] David S. Johnson, Jan Karel Lenstra, and Alexander H. G. Rinnooy Kan. The complexity of the network design problem. *Networks*, 8:279–285, 1978.

[18] Dong-Hee Kim, Jae Dong Noh, and Hawoong Jeong. Scale-free trees: The skeletons of complex networks. *Physical Review E*, 70(046126), 2004.

[19] Guy Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.

[20] Guy Kortsarz and David Peleg. Generating sparse 2-spanners. *Journal of Algorithms*, 17(2):222–236, 1994.

[21] Dieter Kratsch, Hoàng-Oanh Le, Haiko Müller, Erich Prisner, and Dorothea Wagner. Additive tree spanners. *SIAM Journal on Discrete Mathematics*, 17(2):332–340, 2003.

[22] Arthur L. Liestman and Thomas C. Shermer. Additive graph spanners. *Networks*, 23(4):343–363, 1993.

[23] David Peleg and Alejandro A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

[24] David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. *SIAM Journal on Computing*, 18(4):740–747, 1989.

[25] Erich Prisner. Distance approximating spanning trees. In *Proceedings of the 14th Symposium on Theoretical Aspects of Computer Science (STACS'97)*, volume 1200 of *Lecture Notes in Computer Science*, pages 499–510. Springer, February/March 1997.

[26] Gert Sabidussi. The centrality index of a graph. *Psychometrica*, 31:581–603, 1966.

[27] José Soares. Graph spanners: a survey. *Congressus Numerantium*, 89:225–238, 1992.

[28] G. Venkatesan, Udi Rotics, M. S. Madanlal, Johann A. Makowsky, and C. Pandu Rangan. Restrictions of minimum spanner problems. *Information and Computation*, 136(2):143–164, 1997.