Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Combinatorial network abstraction by trees and distances[☆]

Stefan Eckhardt[a,*], Sven Kosub[a], Moritz G. Maaß[a], Hanjo Täubig[a], Sebastian Wernicke[b]

[a] *Fakultät für Informatik, Technische Universität München, Boltzmannstraße 3, D-85748 Garching, Germany*
[b] *Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany*

ARTICLE INFO

Article history:
Received 13 March 2006
Received in revised form 16 November 2007
Accepted 23 March 2008
Communicated by D. Peleg

Keywords:
Network abstraction
Spanning trees
Matrix norms
NP-completeness
Inapproximability

ABSTRACT

We draw attention to combinatorial network abstraction problems. These are specified by a class $\mathcal{P}$ of pattern graphs and a real-valued similarity measure ϱ that is based on certain graph properties. For a fixed pattern $\mathcal{P}$ and similarity measure ϱ, the optimization task on a given graph G is to find a subgraph $G' \subseteq G$ which belongs to $\mathcal{P}$ and minimizes $\varrho(G, G')$. In this work, we consider this problem for the natural and somewhat general 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 standard vector- and matrix-norms. Complexity analysis within a unifying framework shows that all considered variants of the problem are NP-complete, except for the case of distance-minimization with respect to the norm L_∞. If a subset of edges can be ”forced” into the spanning tree, no polynomial-time constant-factor approximation algorithmexists for the distance-approximation problems unless P = NP.

1. Introduction

1.1. Motivation

Network analysis aims to algorithmically expose 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). These structures might explain network functionality and the inter-relationships between its members on several levels of aggregation. A (simple) sub-network 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 $\mathcal{P}$ 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. Notably, the dual problem of fixing $\varrho(G, G')$ and finding a graph class $\mathcal{P}$ that meets this constraint has been considered extensively in the literature, whereas our approach seemingly has not yet been systematically investigated.

We restrict ourselves to trees as the class of pattern graphs in this work since they are the most sparse and simple subgraphs that may connect all vertices of a network (however, many results easily carry over to related structures such as spanning subgraphs with a restricted number of edges). 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.

[☆] An extended abstract of this work appears in the proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC'05), Springer LNCS 3827, pp. 1100–1109, held in Sanya, PR China, December 19–21, 2005.
[*] Corresponding author. Tel.: +49 89 28917700.
E-mail addresses: eckhardt@in.tum.de (S. Eckhardt), kosub@in.tum.de (S. Kosub), maass@in.tum.de (M.G. Maaß), taeubig@in.tum.de (H. Täubig), wernicke@minet.uni-jena.de (S. Wernicke).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.03.037

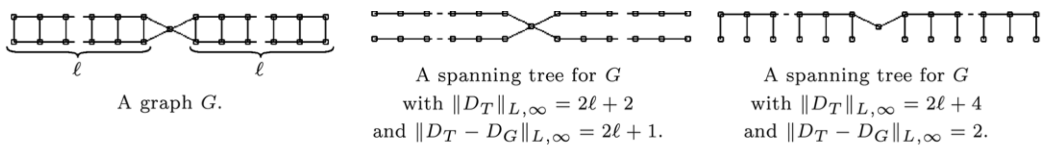
ℓ
ℓ
A graph G.
A spanning tree for G with $\|D_T\|_{L,\infty} = 2\ell + 2$ and $\|D_T - D_G\|_{L,\infty} = 2\ell + 1$.
A spanning tree for G with $\|D_T\|_{L,\infty} = 2\ell + 4$ and $\|D_T - D_G\|_{L,\infty} = 2$.

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

(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. Without essentially changing the network characteristics, this is worthwhile given storage limitations and time requirements.

As to suitable graph properties (i.e., those for which a high amount of similarity between a network and its abstraction is desirable), this work concentrates on distances as an inherent graph property. To quantify this degree of similarity, we use standard vector and matrix norms $\|\cdot\|_r$ (see Section 1.4 for a review and definitions) on the distance matrix D_G of an input graph $G = (V, E)$ 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) [13,8]. For the L_∞ matrix norm, the tree realizing the minimum is known as the minimum diameter spanning tree [6,11].
(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, using the L_∞ matrix-norm here we would be searching for 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 would again be 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 provide good approximate solutions for each other. An example for this (with respect to L_∞) is given in Fig. 1.

1.2. 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 [6,11] and that it is NP-complete to decide on input (G, γ) whether there is a spanning tree T of G such that $\|D_T\|_{L,1} \leq \gamma$ [13]. 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] We close this gap here.

- In Section 3, 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 of this problem (as, e.g., in [5]), meaning that problem instances may 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 the minimum diameter spanning tree problem becomes NP-complete.
- In Section 4, 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 polynomial-time-approximated within a constant factor unless P = NP.
- Finally in Section 5, 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.

[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.

1.3. Related work

In addition to the already mentioned minimum diameter spanning trees [6,11] and MAD-trees [13,8], there are several notions of distance-approximability by trees that 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 the stretch is at most γ, then the tree is called γ-multiplicative tree spanner (see, e.g., [22]). Finding a minimum maximum-stretch tree is NP-hard even for unweighted planar graphs [10], and cannot be approximated by a factor better than $(1+\sqrt{5})/2$ unless P = NP [19]. The problem of finding the minimum *average*-stretch tree is also NP-hard [13]. Recently, combinations of additive and multiplicative tree-spanners have been proposed [9].

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 $u, v \in V(G)$. 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 then is to find an $f(x)$-spanner with the minimum number of edges, a problem somewhat 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 [20,5,4,16]. The version of this problem that is probably the closest to the problems we consider here is to ask for a given graph G and two given parameters m, t if there exists is 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, the problem considered in [18] becomes the problem of finding the best possible distance-approximating spanning tree with respect to $\|\cdot\|_{L,\infty}$. However, their NP-completeness proof 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.

1.4. Notation

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

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

Trivially, for symmetric matrices we have $\|A\|_1 = \|A\|_\infty$. Thus, all our results regarding the maximum column sum norm also hold for the maximum row sum norm (and vice versa). Therefore, we only consider the maximum column sum norm in our results to avoid redundancy.

2. Gadgets

2.1. Graph representation of X3C *instances*

The NP-complete Exact-3-Cover problem is defined as follows:

Exact-3-Cover (X3C)
Input: A family $\mathcal{C} = \{C_1, \dots, C_s\}$ of 3-element subsets of a set $L = \{l_1, \dots, l_{3m}\}$.
Question: Does there exist a subfamily $\mathcal{S} \subseteq \mathcal{C}$ of pairwise disjoint sets such that $\bigcup_{A\in\mathcal{S}} = L$?

A subfamily $\mathcal{S}$ satisfying the required properties is called an *admissible solution* to an instance $(\mathcal{C}, L)$. Suppose we are given an X3C instance $(\mathcal{C}, L)$. Let a and b be arbitrary natural numbers. Following a construction from [13], we define a graph $G_{a,b}(\mathcal{C}, L)$ illustrated in Fig. 2. It consists of the vertex set

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

[2] 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 $\|x\|_p \overset{\text{def}}{=} \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ for $1 \leq p < \infty$.

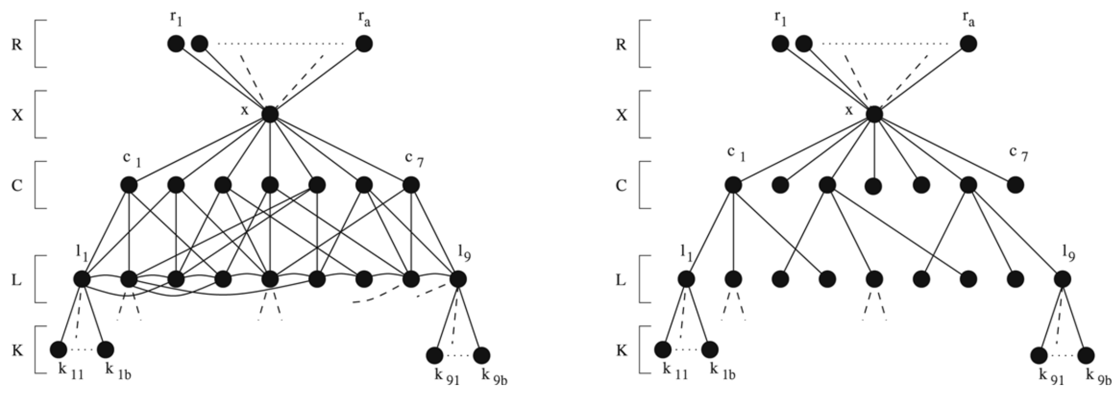

Fig. 2. Graph representation of an X3C instance and a corresponding solution tree.

Since the image somewhat resembles a helicopter, for the sake of readability we identify the vertices from the different sets as *rotor* (vertices from R), *axis* (vertex x), *upper body* (vertices from C), *lower body* (vertices from L) and *skid vertices* (vertices from K). The edge set is defined as follows

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

As with the vertices, we give names to the different groups of edges:

- For all $\mu \in \{1,\ldots,a\}$, the edge $\{r_\mu, x\}$ is called a *rotor edge*.
- For all $\mu \in \{1,\ldots,3m\}$ and $\nu \in \{1,\ldots,b\}$, the edge $\{k_{\mu,\nu}, l_\mu\}$ is called a *skid edge*.
- For all $\mu \in \{1,\ldots,s\}$ the edge $\{C_\mu, x\}$ is called an *axis edge*.
- For all $\mu \in \{1,\ldots,s\}$ and $\nu \in \{1,\ldots,3m\}$ with $l_\nu \in C_\mu$, the edge $\{C_\mu, l_\nu\}$ is called a *central edge*.
- For all $\mu, \nu \in \{1,\ldots,3m\}$, the edge $\{l_\mu, l_\nu\}$ is called *hull edge*.

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

(1) *All rotor edges and all skid edges are in the tree.*
(2) *If for some upper body vertex C_μ, T does not contain the axis edge $\{C_\mu, x\}$, then we have $d_T(C_\mu, r_\nu) \geq 4$ and $d_T(C_\mu, r_\nu) \geq d_{G_{a,b}(\mathcal{C},L)}(C_\mu, r_\nu) + 2$ for all rotor vertices r_ν.*
(3) *If for some lower body vertex l_μ there is no adjacent central edge in T, then for every rotor vertex r_κ and we have $d_T(l_\mu, r_\kappa) \geq 4$ and $d_T(l_\mu, r_\kappa) \geq d_{G_{a,b}(\mathcal{C},L)}(l_\mu, r_\kappa) + 1$, and for every skid vertex $k_{\mu,\lambda}$ adjacent to l_μ, we have $d_T(k_{\mu,\lambda}, r_\kappa) \geq 5$, and $d_T(k_{\mu,\lambda}, r_\kappa) \geq d_{G_{a,b}(\mathcal{C},L)}(k_{\mu,\lambda}, r_\kappa) + 1$.*

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 consisting of *all* rotor edges, *all* axis edges, *all* skid edges and, for all $C_\nu \in \mathcal{S}$ and $l_\mu \in C_\nu$, the central edge $\{C_\nu, l_\mu\}$.

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, and let $T_\mathcal{S}$ be a solution tree in $G_{a,b}(\mathcal{C}, L)$ that corresponds to $\mathcal{S}$.*

(1) *Every upper body vertex C_μ has degree one or four in $T_\mathcal{S}$.*
(2) *Let u be the axis vertex or a rotor vertex and $v \in V$. Then $d_{T_\mathcal{S}}(u, v) = d_{G_{a,b}(\mathcal{C},L)}(u, v)$.*
(3) *For any two upper body vertices C_μ, C_ν, $d_{T_\mathcal{S}}(C_\mu, C_\nu) = d_{G_{a,b}(\mathcal{C},L)}(C_\mu, C_\nu)$.*

Lemma 3. *Let $(\mathcal{C}, L)$ be an* X3C *instance, $a, b \in \mathbb{N}$, and 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) *The tree T contains all axis edges.*
(2) *For every lower body vertex l_μ, there is an upper body vertex C_ν such that T contains the central edge $\{l_\mu, C_\nu\}$.*
(3) *An upper body 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 every two lower body vertices l_μ, l_ν (with $\mu \neq \nu$), there exist upper body vertices C_κ, C_λ such that the path $(l_\mu, \mathcal{C}_\kappa, x, \mathcal{C}_\lambda, l_\nu)$ exists in T. Thus the hull edge $\{l_\mu, l_\nu\}$ cannot 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. □

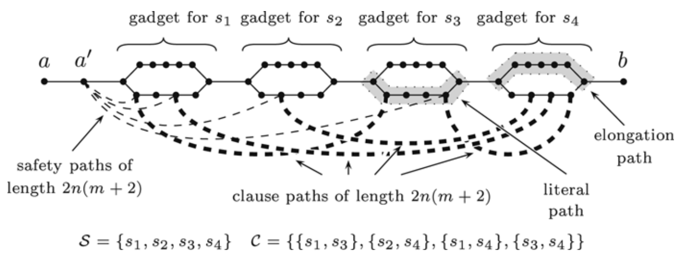
gadget for s_1
gadget for s_2
gadget for s_3
gadget for s_4
a
a'
b
elongation path
safety paths of length $2n(m+2)$
clause paths of length $2n(m+2)$
literal path
$\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ $\mathcal{C} = \{\{s_1, s_3\}, \{s_2, s_4\}, \{s_1, s_4\}, \{s_3, s_4\}\}$

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.

2.2. *Graph representation of* 2HS *instances*

The NP-complete 2-Hitting Set problem is defined as follows[3]:

2-Hitting Set (2HS)
Input: A family $\mathcal{C}$ of m 2-element subsets of a set $\mathcal{S} = \{s_1, \ldots, s_n\}$ and $k \in \mathbb{N}$.
Question: Does there exist a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $\|\mathcal{S}'\| \leq k$ and for each $\mu \in \{1, \ldots, m\}$, there is at least one element in $C_\mu \cap \mathcal{S}'$?

A subset $\mathcal{S}' \subseteq S$ having the required properties 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:

- the three vertices a, a', and b.
- for each $s_\mu \in \mathcal{S}$, 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*. Each such subgraph is called a *literal gadget* G_μ.
- for each clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$, a path of length $2n(m+2)$, called *clause path*, connecting v_μ^ν with v_μ^κ and a path of length $2n(m + 2)$, called *safety path*, connecting v_μ^ν with a'.

The construction is illustrated in Fig. 3.

In each spanning tree T of the gadget, all cycles must be broken. The main idea behind the gadget is that the literal paths are tuned such that they are exactly one edge longer than the elongation paths. If we restrict ourselves to *consider only the subset of all spanning trees which contain all edges of all clause paths*, then the cycles induced by the clause paths can only be broken by destroying a certain amount of literal paths, each such destruction causes the distance between a and b to increase by exactly one. This increase is called the *penalty*. The following lemma formalized this idea.

Lemma 4. *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 $d_T(a, b) \leq d_{G(\mathcal{C},\mathcal{S},k)}(a, b) + k$.*

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'.[4]
(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 $\{\{v_{\mu-1}^\nu, v_\mu^\nu\}, \{v_m^\nu, v'_\nu\}\}$ and $\{\{v_{\mu-1}^\kappa, v_\mu^\kappa\}, \{v_m^\kappa, v'_\kappa\}\}$ is removed, thus either v_μ^ν or v_μ^κ 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 $\{\{v_{\mu-1}^\nu, v_\mu^\nu\}, \{v_m^\nu, v'_\nu\}\}$ and $\{\{v_{\mu-1}^\kappa, v_\mu^\kappa\}, \{v_m^\kappa, v'_\kappa\}\}$ are removed, in which case neither v_μ^ν nor v_μ^κ is reachable via the clause path from any of $v_\nu, v'_\nu, v_\kappa, v'_\kappa$. A cycle induced by

[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.

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

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,

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

($\Leftarrow$) Suppose T is a spanning tree of $G(\mathcal{C},\mathcal{S},k)$ containing all edges of the clause paths and satisfying $d_T(a,b) \le d_{G(\mathcal{C},\mathcal{S},k)}(a,b)+k$. Since $d_{G(\mathcal{C},\mathcal{S},k)}(a,b)+k \le 2+n(m+3) < 2+2n(m+2)$, 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., $(v_\mu, v_1^\mu, \ldots, v_m^\mu, v'_\mu)$ is a path in T. Assume we have for any clause $C_\mu = \{s_\nu, s_\kappa\} \in \mathcal{C}$ (where $\nu < \kappa$) that $C_\mu \cap \mathcal{S}' = \emptyset$. The clause path connects v_μ^ν with v_μ^κ. Since $s_\nu, s_\kappa \notin \mathcal{S}'$, the vertex v_μ^ν is connected to v'_ν is connected to v^κ is connected to v_μ^κ which is a contradiction to T being a tree. $\square$

3. 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: DMST (with respect to $\|\cdot\|_r$)
Input: A connected graph G and an algebraic number γ
Question: Does G contain a spanning tree T with $\|D_T\|_r \le \gamma$?

We also consider the *fixed-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 $\|D_T\|_r \le \gamma$.

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 $1 \le p < \infty$. Note that the case $p = 1$ corresponds to the MAD-tree problem which was shown to be NP-complete in [13]. In fact, our proof generalizes the X3C-based proof technique from [13].

Theorem 5. DMST *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 $G_{a,0}(\mathcal{C},L)$ for any X3C instance $(\mathcal{C},L)$. We will fix the parameters a 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 $G_{a,0}(\mathcal{C},L)$ has a spanning tree T such that $\|D_T\|_{L,p}^p \le \gamma^p$.

Suppose $\mathcal{S} \subseteq \mathcal{C}$ is an admissible solution to instance $(\mathcal{C},L)$. Let $T_{\mathcal{S}}$ be the corresponding spanning tree of $G_{a,0}(\mathcal{C},L)$. Define:

$$N \stackrel{\text{def}}{=} \sum_{\substack{u,v\in V \text{ and} \\ u\in R\cup X \text{ or } v\in R\cup X}} d_{G_{a,0}(\mathcal{C},L)}(u,v)^p \text{ and } M \stackrel{\text{def}}{=} \sum_{u,v\in C\cup L} d_{T_{\mathcal{S}}}(u,v).$$

By Proposition 2, N remains unchanged if the distances in $G_{a,0}(\mathcal{C},L)$ are replaced by the distances in $T_{\mathcal{S}}$. We now set our parameters as follows:

$$a \stackrel{\text{def}}{=} \left\lceil \frac{M}{4^p - 3^p} \right\rceil \text{ and } \gamma \stackrel{\text{def}}{=} (N+M)^{1/p}.$$

Clearly, $\|D_{T_{\mathcal{S}}}\|_{L,p}^p = N + M = \gamma^p$. Thus, $T_{\mathcal{S}}$ is a spanning tree of $G_{a,0}(\mathcal{C},L)$ having the desired distance property.

Suppose T is a spanning tree of $G_{a,0}(\mathcal{C},L)$ such that $\|D_T\|_{L,p}^p \le \gamma^p$. We apply the characterization of a solution tree specified in Lemma 3 and show by contradiction 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 some axis edge $\{C_\mu, x\}$ is not in T. Then, for the corresponding upper body vertex C_μ, $d_T(C_\mu, x) \ge 3$ and for all rotor vertices r_ν, $d_T(C\mu, r_\nu) \ge 4$. Thus, $\|D_T\|_{L,p}^p \ge N - 1^p - 2^p a + 3^p + 4^p a$ and we conclude

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

  a contradiction.
- Assume to the contrary that there is some lower body vertex l_μ not adjacent to any upper body vertex $\mathcal{C}_\nu$ in T. Then, $\|D_T\|_{L,p}^p \ge N - 2^p - 3^p a + 3^p + 4^p a$ and we conclude

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

  a contradiction.

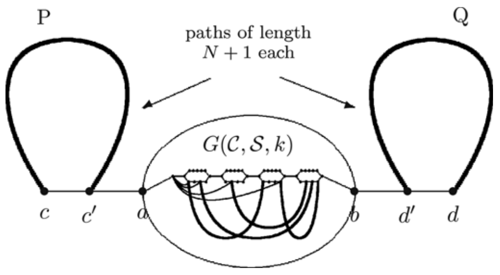

Fig. 4. Construction for proving the NP-completeness of the fixed-edge version of DMST with respect to $\|\cdot\|_{L,\infty}$. The dashed bold paths consist solely of edges that are included in the edge set required to be in a spanning tree for G).

- Note that, if all axis edges are in T and for every lower body vertex, there is at least one adjacent upper body vertex in T, then all edges but the central edges are already fixed by now and the distances in T and $G_{a,0}(\mathcal{C},L)$ 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 central 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

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

a contradiction.

This proves the theorem by applying Lemma 3. □

We know from the literature [14,6,11] that a minimum diameter spanning tree in a graph can be found in time $O(mn+n^2\log n)$. However, if we require that certain edges have to be in the spanning tree, feasibility becomes out of reach.

Theorem 6. *The fixed-edge version of* DMST *with respect to* $\|\cdot\|_{L,\infty}$ *is* NP-*complete.*

Proof. Containment in NP is obvious. To show the NP-hardness, we describe a reduction from 2HS based on the graph representation $G(\mathcal{C},\mathcal{S},k)$ for any given instance $(\mathcal{C},\mathcal{S},k)$ of 2HS. 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. 4.

Define the set E_0 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 $E_0\subseteq E(T)$ must include P and Q completely. Therefore, $d_T(c,d)\ge 2N$. 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 $d_T(v,w)\le N-1$ and $d_T(v,u)\le 2N-1$. 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 $\gamma \overset{\text{def}}{=} 2N+4+n(m+2)+k$ where $n=\|\mathcal{S}\|$ and $m=\|\mathcal{C}\|$ for a given 2HS 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_0 and satisfies $\|D_T\|_{L,\infty}\le\gamma$. This proves the theorem. □

As the next theorem shows, distance-minimizing spanning trees are hard to find under the maximum column sum norm and, thus, the maximum row sum norm as well.

Theorem 7. DMST *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 $G_{a,0}(\mathcal{C},L)$ for a given X3C instance $(\mathcal{C},L)$ and an appropriate choice of the parameters a 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 $G_{a,0}(\mathcal{C},L)$ has a spanning tree T such that $\|D_T\|_1\le\gamma$.

Suppose $\mathcal{S} \subseteq \mathcal{C}$ is an admissible solution to $(\mathcal{C}, L)$. Let $T_{\mathcal{S}}$ be the corresponding spanning tree in the graph $G_{a,0}(\mathcal{C}, L)$. The vertices in the rotor, axis or lower body sets of vertices all have the same column sums. We calculate for a rotor vertex r_μ and an lower body vertex l_ν:

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

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

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

We define our parameters as follows:

$$a \overset{\text{def}}{=} s + 12m + 8 \quad \text{and} \quad \gamma \overset{\text{def}}{=} 6s + 48m + 16.$$

Clearly, we have $\|D_{T_{\mathcal{S}}}\|_1 = 6s + 48m + 16 = \gamma$. Thus, $T_{\mathcal{S}}$ is a spanning tree in $G_{a,0}(\mathcal{C}, L)$ having a distance property as desired.

Suppose T is a spanning tree in $G_{a,0}(\mathcal{C}, L)$ satisfying $\|D_T\|_1 \leq \gamma$. We apply the characterization of a solution tree given in Lemma 3 and show by contradiction that all conditions are satisfied. The structure of the rest of the proof is very similar to the structure of the proof of Theorem 5, except that we must check the conditions with different parameters: First, if for some upper body vertex C_μ the edge $\{C_\mu, x\}$ does not belong to T, then $\sum_{v \in V} d_T(C_\mu, v) \geq 4a + 2s + 6m - 5 = 6s + 54m + 27 > \gamma$, a contradiction. Second, if there is a lower body vertex l_μ not adjacent to any vertex upper body vertex C_ν in T, then $\sum_{v \in V} d_T(l_\mu, v) \geq 4a + 2s + 3m + 2 = 6s + 51m + 34 > \gamma$, a contradiction. Finally, assume to the contrary that there is a vertex upper body vertex C_μ having two or three neighbors in T. Remember that the first two conditions imply that there is no hull edge $\{l_\mu, l_\nu\}$ in T. Let l_ν be one of C_μ's neighbors in T. We calculate $\sum_{v \in V} d_T(l_\nu, v) \geq 3a + 3s + 12m - 5 = 6s + 48m + 19 > \gamma$, a contradiction. This proves the theorem by Lemma 3. □

Remark 8. Both constructions in Theorems 5 and 7 do not rely on using 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). This means that computing distance-minimizing spanning trees for these norms is already NP-hardin planar graphs.

4. Trees that approximate distances

We now turn to the problem of finding spanning trees approximating the distances in a graph reasonably well under a certain given matrix norm.

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

We also examine the fixed-edge version of this problem, which is specified in the same way as for DMST.

4.1. NP-completeness results

Independent of the norm, we show all problems to be NP-complete. Notice that with respect to the L_1 matrix-norm, computing distance-minimizing and optimal distance-approximating spanning trees is equivalent. An immediate consequence is NP-completeness under L_1 matrix-norm (from Theorem 5 or [13]).

Theorem 9. DAST *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 $G_{a,0}(\mathcal{C}, L)$ for a given X3C instance $(\mathcal{C}, L)$ and an appropriate choice of the parameters a 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 $G_{a,0}(\mathcal{C}, L)$ has a spanning tree T such that $\|D_{G_{a,0}(\mathcal{C},L)} - D_T\|_{L,p}^p \leq \gamma^p$.

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

$$\|D_{G_{a,0}(\mathcal{C},L)} - D_{T_{\mathcal{S}}}\|_{L,p}^p = 2\left(\underbrace{2^p 3(s-m) + 3m(s-1)}_{\text{upper to lower body}} + \underbrace{3m + 3^p \frac{9m^2 - 9m}{2}}_{\text{lower to lower body}}\right).$$

We now set our parameters as follows:

$$a \overset{\text{def}}{=} \gamma^p \text{ and } \gamma \overset{\text{def}}{=} \|D_{G_{a,0}(\mathcal{C},L)} - D_{T_{\mathcal{S}}}\|_{L,p}.$$

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 $G_{a,0}(\mathcal{C}, L)$ having the desired distance property.

Suppose T is a spanning tree in $G_{a,0}(\mathcal{C}, L)$ satisfying $\|D_{G_{a,0}(\mathcal{C},L)} - D_T\|_{L,p}^p \leq \gamma^p$. We apply the characterization of a solution tree given in Lemma 3 and show by contradiction that all conditions are satisfied. Again, the structure of the proof is very similar to the proof Theorem 5, but this time we used the parameter a to control the increase in distance, not the distance itself. First, if for some upper body vertex C_μ, the edge axis edge $\{C_\mu, x\}$ does not belong to T, this implies $d_T(C_\mu, v) \geq d_{G_{a,0}(\mathcal{C},L)}(C_\mu, v) + 2$, for v a rotor or axis vertex. Thus, $\|D_{G_{a,0}(\mathcal{C},L)} - D_T\|_{L,p}^p \geq (a+1)2^p > \gamma^p$, a contradiction. Second, if there is a lower body vertex l_μ not adjacent to any upper body vertex C_ν in T, then $d_T(l_\mu, v) \geq d_{G_{a,0}(\mathcal{C},L)}(l_\mu, v) + 1$, for v a rotor or axis vertex. This gives $\|D_{G_{a,0}(\mathcal{C},L)} - D_T\|_{L,p}^p \geq a + 1 > \gamma^p$, a contradiction. Finally, note that, if the first and second condition in Lemma 3 are both satisfied, then all edges but the central edges are already fixed by now and the distances in T and $G_{a,0}(\mathcal{C}, L)$ are the same. For the distances from upper to lower body vertices we have:

$$d_T(l_\mu, C_\nu) = \begin{cases} d_{G_{a,0}(\mathcal{C},L)}(l_\mu, C_\nu) & \text{if edge } \{l_\mu, C_\nu\} \text{ is in } T \\ d_{G_{a,0}(\mathcal{C},L)}(l_\mu, C_\nu) + 1 & \text{if edge } \{l_\mu, C_\nu\} \text{ is not in } T \text{ and } l_\mu \notin C_\nu \\ d_{G_{a,0}(\mathcal{C},L)}(l_\mu, C_\nu) + 2 & \text{if edge } \{l_\mu, C_\nu\} \text{ is not in } T \text{ and } l_\mu \in C_\nu. \end{cases}$$

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

$$\sum_{\mu=1}^{3m}\sum_{\nu=1}^{s}\left(d_{G_{a,0}(\mathcal{C},L)}(l_\mu, C_\nu) - d_T(l_\mu, C_\nu)\right)^p = 3s(m-1) + 2^p(3h_1 + 2h_2 + h_3).$$

Note that $3h_1 + 2h_2 + h_3 = 3(s - m)$. For the distances between lower body vertices l_μ, l_ν, we obtain for $\mu \neq \nu$,

$$d_T(l_\mu, l_\nu) = \begin{cases} d_{G_{a,0}(\mathcal{C},L)}(l_\mu, l_\nu) + 1 & \text{if edges } \{l_\mu, C_\kappa\} \text{ and } \{l_\nu, C_\kappa\} \text{ belong to } T \\ & \text{for some } \kappa \in \{1, \ldots, s\} \\ d_{G_{a,0}(\mathcal{C},L)}(l_\mu, l_\nu) + 3 & \text{otherwise.} \end{cases}$$

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

$$\sum_{\mu=1}^{3m}\sum_{\nu=1}^{3m}\left(d_{G_{a,0}(\mathcal{C},L)}(l_\mu, l_\nu) - d_T(l_\mu, l_\nu)\right)^p = g + 3^p(9m^2 - 3m - g).$$

The maximum possible value of g is $6m$. The case $g = 6m$ implies $h_4 = m$ and therefore corresponds to the case that the third condition in Lemma 3 is satisfied. Assume to the contrary $g < 6m$. Then we have

$$\begin{aligned} \|D_{G_{a,0}(\mathcal{C},L)} - D_T\|_{L,p}^p - \gamma^p &\geq 6s(m-1) + 2^{p+1}(3s - 3m) + g + 3^p(9m^2 - 3m - g) \\ &\quad - 6s(m-1) - 2^{p+1}(3s - 3m) - 6m - 3^p(9m^2 - 9m) \\ &= (3^p - 1)(6m - g) > 1, \end{aligned}$$

a contradiction. This proves the theorem by Lemma 3. □

To be able to carry out the NP-completeness proof for the L_∞ matrix-norm, it is helpful to first have a completeness result for the fixed-edge version.

Lemma 10. *The fixed-edge version of* DAST *with respect to* $\|\cdot\|_{L,\infty}$ *is* NP-*complete.*

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

We now try to get rid of the fixed edges. In order to achieve this, we replace fixed edges 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. 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.

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

Proof. For any spanning tree T of a graph $G = (V, E)$, we define $\delta_T(v) \overset{\text{def}}{=} \max_{w\in V}(d_T(v, w) - d_G(v, w))$. Let P be the path $(v, u_1, \dots, u_k, w)$ 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 distancechanges 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 $\mu \in \{1, \dots, \lfloor \frac{k}{2} \rfloor\}$ 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 $\mu \in \{\lceil \frac{k+1}{2} \rceil, \dots, k\}$ 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 $B_{=k}(z) \overset{\text{def}}{=} \{x \in V \mid d_T(x, z) - d_G(x, z) = k\}$. 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:

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

  For $k > 4$ this leads to a contradiction and thus, our claim is true in this case. The case $k = 4$ 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 $d_{T'}(u_\mu, y) \le d_{G'}(u_\mu, y) + k$.
  - For $\mu \in \{1, \dots, \lfloor \frac{k}{2} \rfloor\}$ and $y \in V(G)$, the shortest path between u_μ and y visits v. Thus, $d_{T'}(u_\mu, y) \le d_{G'}(u_\mu, y) + k$.
  - For $\mu \in \{\lceil \frac{k+1}{2} \rceil + 1, \dots, k\}$ and $y \in V(G)$, the shortest path between u_μ and y visits w. Thus, $d_{T'}(u_\mu, y) \le d_{G'}(u_\mu, y) + k$.
  - 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, $d_{T'}(u_\mu, y) \le d_{G'}(u_\mu, y) + k$. For $y \notin B_{=k}(v) \cup B_{=k}(w)$ we obtain

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

  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, $d_{T'}(u_\mu, y) - d_{G'}(u_\mu, y) = \min(d_T(v, y) - d_G(v, y), d_T(w, y) - d_G(w, y)) \le k$.
  This completes the second case.

($\Leftarrow$) Suppose there is a spanning tree T' for G' with $\|D_{T'} - D_{G'}\|_{L,\infty} \le k$. 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. 5 for illustration):

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

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

  a contradiction.

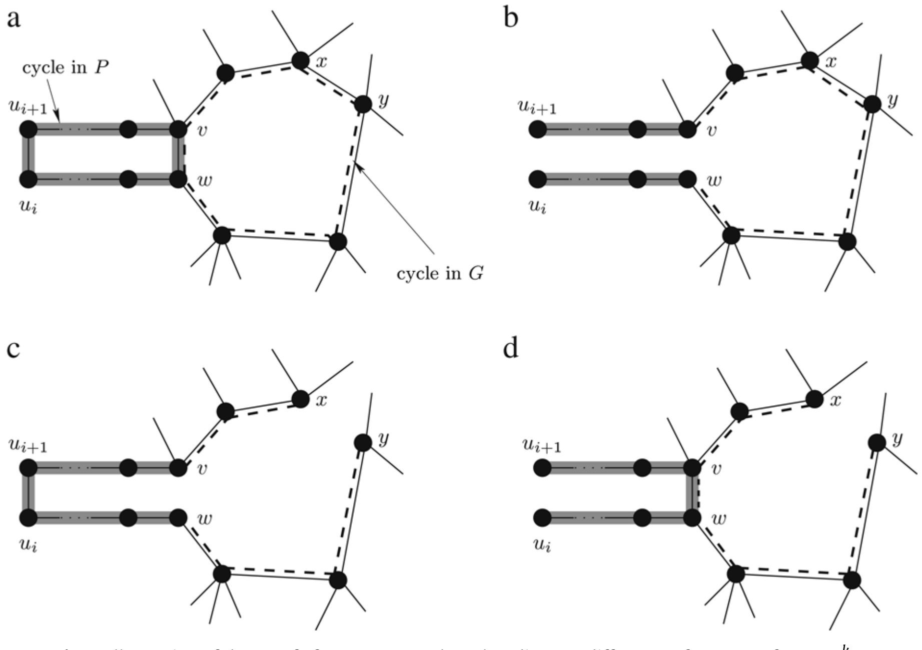

Fig. 5. Illustration of the proof of Lemma 11. Only T_3 has distance difference of at most k for $i = \lfloor \frac{k}{2} \rfloor$.

- 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

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

  again a contradiction.

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. DAST *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 DAST. We may restrict ourselves to instance (G, E_0, γ) with $\gamma > 3$. First, note that if E_0 contains any bridges, we may remove these from E_0 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_0 form a cycle, we may immediately reject the instance. Using Lemma 11 we describe the following reduction: for every edge $\{v, w\} \in E_0$ iteratively add a path $(v, u_1, \ldots, u_k, w)$ with new vertices. Let G' be the resulting graph, which of course can be constructed in polynomial time in the size of G. An easy induction on the size of E_0 now shows that G has a spanning tree containing all edges of E_0 such that $\|D_T - D_G\|_{L,\infty} \leq \gamma$ if and only if G' has a spanning tree T' such that $\|D_{T'} - D_{G'}\|_{L,\infty} \leq \gamma$. $\square$

Concluding this section, we show the hardness with respect to the maximum column sum norm.

Theorem 13. DAST *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 $\mu \in \{1, \ldots, 3m\}$, 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 $h_{max} \overset{\text{def}}{=} \max_\mu h(\mu)$.

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. Thus, for a rotor vertex r_μ, the axis vertex x, a lower body vertex l_ν

and a skid vertex $k_{\nu,\kappa}$, we calculate:

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

For upper body vertices we have to make a distinction between vertices with one neighbor in T and vertices with four neighbors in T. We obtain for an upper body vertex C_μ:

$$\sum_{v\in V} d_T(C_\mu, v) - d_{G_{a,b}(\mathcal{C},L)}(C_\mu, v) = \begin{cases}(3m+3)(b+1) & \text{if } C_\mu \text{ has 1 neighbor in } T_{\mathcal{S}}\\ (3m-3)(b+1) & \text{if } C_\mu \text{ has 4 neighbors in } T_{\mathcal{S}}.\end{cases}$$

Setting the parameters $a \overset{\text{def}}{=} \gamma + 1$, $b \overset{\text{def}}{=} s + 1$ and $\gamma \overset{\text{def}}{=} s + h_{\max} + 9m - 9 + b(9m - 7)$, we get $\|D_{G_{a,b}(\mathcal{C},L)} - D_T\|_1 = s + h_{\max} + 9m - 9 + b(9m - 7) = \gamma$. 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 $\|D_{G_{a,b}(\mathcal{C},L)} - D_T\|_1 \le \gamma$. We apply the characterization of a solution tree given in Lemma 3 and show by contradiction, that all conditions are satisfied.
First, assume to the contrary some axis edge $\{C_\mu, x\}$ does not belong to T. Then $\sum_{v\in V} d_T(C_\mu, v) - d_{G_{a,b}(\mathcal{C},L)}(C_\mu, v) \ge 4a > \gamma$, a contradiction. Second, assume to the contrary that there is a lower body vertex l_μ not adjacent to any upper body vertex C_ν in T. Then $\sum_{v\in V} d_T(l_\mu, v) - d_{G_{a,b}(\mathcal{C},L)}(l_\mu, v) \ge a > \gamma$, a contradiction. Finally note that, if the first and second condition in Lemma 3 are both satisfied, then all edges but the central edges 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 there is an upper body 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

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

a contradiction. This proves the theorem by Lemma 3. □

4.2. Inapproximability results

In this section we show that, independent of the norm used, it is hard to approximate the fixed-edge version of DAST in polynomial time within a constant factor. To put this into more precise terms, for a given graph G and a set $E_0 \subseteq E(G)$ of fixed-edges we say that an algorithm $\mathcal{A}$ is a *constant-factor approximation algorithm* for the fixed-edge version of DAST with respect to $\|\cdot\|$ if $\mathcal{A}$ in polynomial time computes a spanning tree $T_{\mathcal{A}}$ of G with $E_0 \subseteq E(T)$ such that $\|D_{T_{\mathcal{A}}} - D_G\| \le \delta \cdot \|D_{T_{\text{opt}}} - D_G\|$ for some constant $\delta > 0$. Here, T_{opt} is the *optimal tree*, i.e., $\|D_{T_{\text{opt}}} - D_G\| = \min_T \|D_T - D_G\|$ over all spanning trees T of G.

The inapproximability proofs are based on the 2HS graph representation. To sketch the main idea of our construction: In Lemma 4, it was shown that the size of the solution of the 2HS instance equals the number of literal paths in the graph representation that are opened in the corresponding spanning tree T. Futhermore, for each literal path that is open in T the distance between a and b is increased by one, because the shortest path in T connecting a and b must use an elongation path instead of the shorter literal path. In the following, we refer to this as the *penalty* for opening the literal path. Clearly, in the above construction, the penalty for each opened literal path is one. The main idea is to find a graph representation such that the penalty is increased. More formally, the *refined* graph representation $G(\mathcal{C}, \mathcal{S}, k, j)$ of an instance $(\mathcal{C}, \mathcal{S}, k)$ of 2HS depends on an additional constant factor j and is constructed *recursively* such that it has a spanning tree T containing a set of fixed edges $E_0 \subseteq E(T)$ with $d_T(a, b) \le d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) + k^j$ if and only if the instance $(\mathcal{C}, \mathcal{S}, k)$ has a solution of size k.

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 the same as the graph $G(\mathcal{C}, \mathcal{S}, k)$. For $j > 1$, define $l_{j-1} \overset{\text{def}}{=} d_{G(\mathcal{C},\mathcal{S},k,j-1)}(a, b)$. Then $G(\mathcal{C}, \mathcal{S}, k, j)$ consists of

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

For each clause $\mu \in \mathcal{C}$ the vertices a_μ and b_μ are connected via the literal path $(a_\mu, v_1^\mu, \ldots, v_{l_{j-1}-1}^\mu, b_\mu)$ of length l_{j-1}. Furthermore the edges $\{a, a'\}$,$\{a', a_1\}$, $\{b_m, b\}$, and the edges $\{b_i, a_{i+1}\}$ for all $1 \le i \le m-1$ are in $G(\mathcal{C}, \mathcal{S}, k, j)$. Again, the graph size is polynomial in the size of the instance $(\mathcal{C}, \mathcal{S}, k)$ and j. The following Lemma is the equivalent to Lemma 4 for the original graph representation of 2HS:

Lemma 14. *Let $(\mathcal{C}, \mathcal{S}, k)$ be an instance of* 2HS *and $j \in \mathbb{N}_+$.*

1. *We have $d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) = 3\frac{n^{j+1}-1}{n-1} + n^j(m-1) - 1$.*
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 $G(\mathcal{C}, \mathcal{S}, k, j)$ 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 $d_T(a, b) \le d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) + k^j$.*

Proof. For $j = 1$ the statements equal those of Lemma 4. Suppose $j > 1$, and assume that the lemma holds for $G(\mathcal{C}, \mathcal{S}, k, j-1)$. Again, let $l_{j-1} = d_{G(\mathcal{C},\mathcal{S},k,j-1)}(a, b)$. For the first statement note that the path between a and b using a' and all literal paths P_μ for all $\mu \in \mathcal{C}$ has length $2 + n \cdot (l_{j-1} + 1)$. On the other hand, a path connecting a and b using either a clause path or a safety path has length at least $2 \cdot n \cdot l_{j-1}$ and thus does not determine the distance between a and b. Using an elongation gadget instead of a literal path yields the same distance. Thus we have proven that

$$\begin{aligned} d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) &= 2 + n \cdot (l_{j-1} + 1) = 2 + n \cdot \left(3\frac{n^j - 1}{n-1} + n^{j-1}(m-1)\right) \\ &= 3\frac{n^{j+1} - 1}{n-1} + n^j \cdot (m-1) - 1. \end{aligned}$$

Assume that the second statement holds for $G(\mathcal{C}, \mathcal{S}, k, j-1)$. We show that it holds for $G(\mathcal{C}, \mathcal{S}, k, j)$ as well.

($\Rightarrow$) Suppose $\mathcal{S}'$ is an admissible solution to $(\mathcal{C}, \mathcal{S}, k)$, i.e., $\|\mathcal{S}'\| \le k$. 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 $v_0^\nu = a_\nu$ and $v_0^\kappa = a_\kappa$ 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'.[5]
2. For each $s_\mu \in \mathcal{S}'$, remove edge $\{v_{l_{j-1}-1}^\mu, b_\mu\}$ (open the literal path).
3. For each $s_\mu \notin \mathcal{S}'$, remove the edge $\{a_\mu, a'_\mu\}$ (open the elongation gadget).

Note that no edge from a clause path was removed. We ensured that each cycle induced by the literal path and elongation gadget 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 $\{\{v_{\mu-1}^\nu, v_\mu^\nu\}, \{v_{l_{j-1}-1}^\nu, b_\nu\}\}$ and $\{\{v_{\mu-1}^\kappa, v_\mu^\kappa\}, \{v_{l_{j-1}-1}^\kappa, b_\kappa\}\}$ is removed, thus either v_μ^ν or v_μ^κ is not reachable via the clause path from a_ν or b_ν (a_κ or b_κ, respectively). An edge from the safety path is removed, except if both $\{\{v_{\mu-1}^\nu, v_\mu^\nu\}, \{v_{l_{j-1}-1}^\nu, b_\nu\}\}$ and $\{\{v_{\mu-1}^\kappa, v_\mu^\kappa\}, \{v_{l_{j-1}-1}^\kappa, b_\kappa\}\}$ are removed, in which case neither v_μ^ν nor v_μ^κ is reachable via the clause path from any of $a_\nu, b_\nu, a_\kappa, b_\kappa$. 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 gadgets ($s_\mu \in \mathcal{S}'$) or literal ($s_\mu \notin \mathcal{S}'$) paths. The penalty for each elongation gadget used is k^{j-1} by induction hypothesis. Since $\|\mathcal{S}\| = k$ elongation gadgets are used in the spanning tree constructed we have

$$\begin{aligned} d_T(a, b) &= 2 + (n - \|\mathcal{S}\|)(l_{j-1} + 1) + \|\mathcal{S}\|(l_{j-1} + 1 + k^{j-1}) \\ &= d_{G(\mathcal{C},\mathcal{S},k,j)} + \|\mathcal{S}\| \cdot k^{j-1} = d_{G(\mathcal{C},\mathcal{S},k,j)} + k^j. \end{aligned}$$

($\Leftarrow$) Suppose T is a spanning tree of $G(\mathcal{C}, \mathcal{S}, k, j)$ containing all edges of the clause paths and satisfying $d_T(a, b) \le d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) + k^j$. Since

$$d_{G(\mathcal{C},\mathcal{S},k,j)}(a, b) + k^j = 3\frac{n^{j+1} - 1}{n-1} + n^j(m-1) - 1 + k^j < 2 + 2 \cdot n \cdot l_{j-1},$$

the path cannot lead via any clause or safety paths. Hence, it must lead via literal paths and elongation gadgets only. The length of any (intact) elongation gadget path is $l_{j-1} + k^{j-1}$ by induction hypothesis, the length of any (intact) literal path is l_{j-1}. Therefore, the path from a to b leads over exactly k elongation gadgets. Let $\mathcal{S}'$ be the set of literals s_μ for which the path leads from a_μ to b_μ via an elongation gadget. 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., $(a_\mu, v_1^\mu, \ldots, v_{l_{j-1}-1}^\mu, b_\mu)$ 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_μ^ν with v_μ^κ. Since $s_\nu, s_\kappa \notin \mathcal{S}'$, the vertex v_μ^ν is connected to b_ν is connected to a_κ is connected to v_μ^κ which is a contradiction to T being a tree. □

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

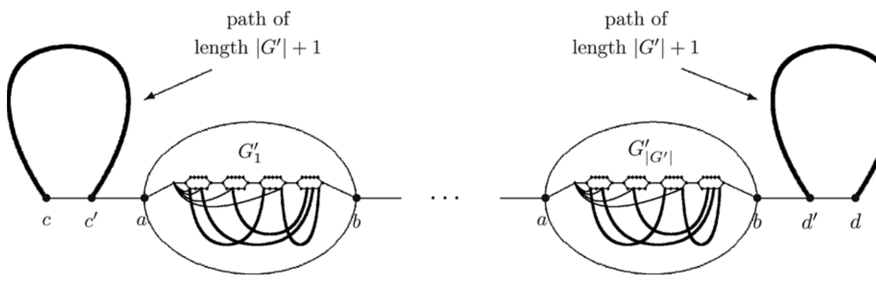

Fig. 6. Illustrating the construction of the graphs used to prove the inapproximability of the fixed-edge version of DAST with respect to $\|\cdot\|_{L,\infty}$. The dashed bold paths consist of fixed edges.

We introduce a shorthand notation. Let $G = (V, E)$ be any graph and let T be any spanning tree of G. We define the mapping $\Delta_G[T] : V \times V \to \mathbb{N}$ for all $u, v \in V$ as $\Delta_G[T](u, v) \overset{\text{def}}{=} d_T(u, v) - d_G(u, v)$.

Lemma 15. *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 $\Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T_{\mathcal{A}}](a, b) \le \delta \cdot \Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T_{\text{opt}}](a, b)$ for any $\delta > 0$.*
Here, T_{opt} is a spanning tree of $G(\mathcal{C}, \mathcal{S}, k, j)$ such that $\Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T_{\text{opt}}](a, b) = \min_T \Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T](a, b)$ where the minimum is taken over all spanning trees T of $G(\mathcal{C}, \mathcal{S}, k, j)$ that include all clause paths.

Proof. Assume there is an algorithm $\mathcal{A}$ that runs in polynomial time such that $\Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T_{\mathcal{A}}](a, b) \le \delta \cdot \Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T_{\text{opt}}](a, b)$ for some $\delta > 0$. Define $j \overset{\text{def}}{=} \lfloor \frac{\log \delta}{\log \frac{8}{7}} \rfloor$. Consider an algorithm $\mathcal{A}'$ that, on instance $(\mathcal{C}, \mathcal{S}, k)$ of 2HS, constructs $G(\mathcal{C}, \mathcal{S}, k, j)$ and applies algorithm $\mathcal{A}$ on $G(\mathcal{C}, \mathcal{S}, k, j)$. Then, by Lemma 14, we have that $(\mathcal{C}, \mathcal{S}, k)$ has a solution of size k_{opt} if and only if there exists a spanning tree T of $G(\mathcal{C}, \mathcal{S}, k, j)$ that contains all edges of all clause paths such that $\Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T] \le k^j_{\text{opt}}$. Thus for the tree $T_{\mathcal{A}}$ computed by the algorithm $\mathcal{A}$, it holds that $\Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T_{\mathcal{A}}](a, b) \le \delta \cdot k^j_{\text{opt}}$. Note that there exists a $k_{\mathcal{A}}$ with $k_{\mathcal{A}} \le \lfloor \sqrt[j]{\Delta_{G(\mathcal{C},\mathcal{S},k,j)}[T_{\mathcal{A}}](a, b)} \rfloor$ if and only if there exists a solution of size $k_{\mathcal{A}}$ to the given 2HS instance. Hence, we conclude

$$k_{\mathcal{A}} \le \delta^{\frac{1}{j}} \cdot k_{\text{opt}} \le \delta^{\frac{\log(8/7)}{\log \delta}} \cdot k_{\text{opt}} \le \frac{8}{7} \cdot k_{\text{opt}}.$$

Therefore, since $\mathcal{A}$ runs in polynomial time, $\mathcal{A}'$ runs in polynomial time (note that j is constant). Thus, we have a polynomial-time algorithm that approximates 2HS within a factor of $\frac{8}{7}$, implying P = NP according to [12]. □

Theorem 16. *Unless* P = NP*, there is no polynomial-time constant-factor approximation algorithm for the fixed-edge version of* DAST *with respect to* $\|\cdot\|_{L,\infty}$.

Proof. Let $(\mathcal{C}, \mathcal{S}, k)$ be a 2HS instance and let $G' = G(\mathcal{C}, \mathcal{S}, k, j)$ be a refined graph representation of $(\mathcal{C}, \mathcal{S}, k)$ for some $j \in \mathbb{N}_+$. Let n denote the number of vertices in G'. We construct a graph G consisting of a chain $G'_1, \ldots G'_n$ of n copies of G' where the vertex b (denoted b_i in G) of G'_i is connected to the vertex a of G'_{i+1} (denoted a_{i+1} in G) by the edge $\{b_i, a_{i+1}\}$ in G. Additionally, G consists of

- four vertices c, c', d', and d connected by edges $\{c, c'\}$ and $\{d, d'\}$,
- two edges $\{c', a_1\}$ and $\{b_n, d'\}$, and
- two paths p and q connecting c, c' and d, d', respectively, each of length $n + 1$.

The construction for G is illustrated in Fig. 6.

Let E'_i be the set of edges in all clause paths in G'_i. Define the set E_0 of fixed edges as

$$E_0 \overset{\text{def}}{=} \bigcup_{i=1}^{n} E'_i \cup E(p) \cup E(q).$$

Then, *any* solution tree T to the fixed-edge version of DAST with respect to $\|\cdot\|_{L,\infty}$ on G must contain p and q completely. However, it must not include $\{c, c'\}$ and $\{d, d'\}$, since otherwise there would be a cycle in the spanning tree. Hence, $\Delta_G[T](d, d') = \Delta_G[T](c, c') = n$. Also, for any vertices $x \in V(G'_1) \cup \cdots \cup V(G'n)$, $y \in G$, and *any* spanning tree T of G,

$$\Delta_G[T](x, y) < \underbrace{n}_{\substack{\text{originating}\\ G'}} + \underbrace{\sum_{i=1}^{n} \Delta_G[T](a_i, b_i)}_{\text{chain}} + \underbrace{n}_{\substack{p,p \text{ or } G'\\ \text{in chain}}} = 2n + \sum_{i=1}^{n} \Delta_G[T](a_i, b_i)$$

while

$$\Delta_G[T](c, d) = 2n + \sum_{i=1}^{n} \Delta_G[T](a_i, b_i). \tag{1}$$

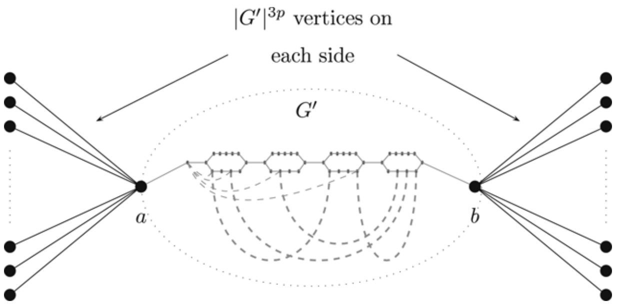

Fig. 7. Illustrating the construction of the graphs used to prove the inapproximability of the fixed-edge version of DAST with respect to $\|\cdot\|_{L,\infty}$. The dashed bold paths consist solely of edges that are included in E_0 (i.e., the set of edges that is required to be in a spanning tree for G).

Hence, for any spanning tree T of G, $\|D_T - D_G\|_{L,\infty} = \Delta_G[T](c,d)$. Furthermore, note that there exists a spanning tree T of G with $E_0 \subseteq E(T)$ such that $\|D_T - D_G\|_{L,\infty} = \Delta_G[T](c,d) = 2n + k \cdot n$ if and only if there exists a spanning tree T' of G'_i with $E'_i \subseteq E(T')$ and $\Delta_{G'}[T'](a,b) = k$ for some $1 \le i \le n$.

In order to prove the inapproximability of the fixed-edge version of DAST with respect to $\|\cdot\|_{L,\infty}$, assume that there exists a polynomial-time constant-factor approximation algorithm $\mathcal{A}$ computing solutions $T_{\mathcal{A}}$. Let $\delta > 0$ be the approximation factor. Again, let T_{opt} be the optimal spanning tree of G with respect to $\|\cdot\|_{L,\infty}$. Note that $d_{T_{\text{opt}}}(a_i,b_i) = d_{T_{\text{opt}}}(a_j,b_j)$ for all $1 \le i,j \le n$. (This follows from the optimality of T_{opt}). Then,

$$\begin{aligned}
0 &\ge \|D_{T_{\mathcal{A}}} - D_G\|_{L,\infty} - \delta \cdot \|D_{T_{\text{opt}}} - D_G\|_{L,\infty} \\
&= \Delta_G[T_{\mathcal{A}}](c,d) - \delta \cdot \Delta_G[T_{\text{opt}}](c,d)) \\
&= 2n + \sum_{i=1}^{n} \Delta_G[T_{\mathcal{A}}](a_i,b_i) - \delta \cdot \Big(2n + \sum_{i=1}^{n} \Delta_G[T_{\text{opt}}](a_i,b_i)\Big) \\
&= 2n + \sum_{i=1}^{n} \Delta_G[T_{\mathcal{A}}](a_i,b_i) - \delta \cdot (2n + n \cdot \Delta_G[T_{\text{opt}}](a_1,b_1)) \\
&\ge 2n + n \cdot \min_{1\le i\le n} \Delta_G[T_{\mathcal{A}}](a_i,b_i) - \delta \cdot (2n + n \cdot \Delta_G[T_{\text{opt}}](a_1,b_1)) \\
&= n \cdot (2 + \min_{1\le i\le n} \Delta_G[T_{\mathcal{A}}](a_i,b_i) - \delta \cdot (2 + \Delta_G[T_{\text{opt}}](a_1,b_1))).
\end{aligned}$$

Without loss of generality, assume that $\Delta_G[T_{\text{opt}}](a_1,b_1) \ge 1$. We obtain

$$\min_{1\le i\le n} \Delta_G[T_{\mathcal{A}}](a_i,b_i) \le 3c \cdot \Delta_G[T_{\text{opt}}](a_1,b_1).$$

Hence, if there were a constant-factor approximation algorithm $\mathcal{A}$ for the fixed-edge version of DAST with respect to $\|\cdot\|_{L,\infty}$, there would also be a polynomial-time algorithm $\mathcal{A}'$ computing a spanning tree $T_{\mathcal{A}'}$ of G' such that $\Delta_{G'}[T_{\mathcal{A}'}](a,b) \le \delta \cdot \Delta_{G'}[T_{\text{opt}}](a,b)$: First construct the chain of copies G and then use $\mathcal{A}$ to compute the tree $T_{\mathcal{A}}$. Choose $T' = T_{\mathcal{A}} \cap G'_i$ such that $\Delta_{G'_i}[T_{\mathcal{A}}](a_i,b_i)$ is minimal. As shown above, this gives a tree $T_{\mathcal{A}'}$ with $\Delta_{G'}[T_{\mathcal{A}'}](a,b) \le \delta \cdot \Delta_{G'}[T_{\text{opt}}](a,b)$. By Lemma 15, this implies P = NP. □

Theorem 17. *Unless* P = NP, *there is no constant-factor approximation algorithm for the fixed-edge version of* DAST *with respect to* $\|\cdot\|_{L,p}$.

Proof. Let $(\mathcal{C},\mathcal{S},k)$ be a 2HS instance and let $G' = G(\mathcal{C},\mathcal{S},k,j)$ be a refined graph representation of $(\mathcal{C},\mathcal{S},k)$ for some $j \in \mathbb{N}_+$. Let n denote the number of vertices in G'. We construct graph G consisting of G' and two groups of n^{3p} vertices each attached by an edge to $a \in V(G')$ and $b \in V(G')$, respectively (see Fig. 7). Observe that all of the added edges are included in a spanning tree T of G. Let the set E_0 of fixed-edges be the set of all edges in all clause paths in G', including the edges in clause paths in $G(\mathcal{C},SF,k,j')$ (with $j' < j$) of which G' is composed.

Let *Inner* be the set of vertices contained in $V(G')$ and $Outer_a$, $Outer_b$ the set of vertices attached to a and b, respectively.[6] Then,

$$\begin{aligned}
\|D_T - D_G\|_{F,p}^p = &\sum_{\substack{u\in Inner\\ v\in Inner}} \Delta_G[T](u,v)^p + 2\cdot \sum_{\substack{u\in Outer_a\\ v\in Inner}} \Delta_G[T](u,v)^p + 2\cdot \sum_{\substack{u\in Outer_b\\ v\in Inner}} \Delta_G[T](u,v)^p + \sum_{\substack{u\in Outer_a\\ v\in Outer_a}} \Delta_G[T](u,v)^p \\
&+ \sum_{\substack{u\in Outer_b\\ v\in Outer_b}} \Delta_G[T](u,v)^p + 2\cdot \sum_{\substack{u\in Outer_a\\ v\in Outer_b}} \Delta_G[T](u,v)^p. \qquad (2)
\end{aligned}$$

[6] Note that $V(G) = Inner \cup Outer_a \cup Outer_b$.

Assume there exists a spanning tree T' of G' with $\Delta_{G'}[T'](a,b) = k$ and $E_0 \subseteq E(T')$. Then there exists a spanning tree T of G where

$$\sum_{\substack{u\in Inner\\ v\in Inner}} \Delta_G[T](u,v)^p \le n^2 \cdot \max_{\substack{u\in Inner\\ v\in Inner}} \Delta_{G'}[T'](u,v)^p \le n^{2+p}, \tag{3}$$

$$\sum_{\substack{u\in Outer_a\\ v\in Inner}} \Delta_G[T](u,v)^p \le n^{3p}\cdot n\cdot n^p, \tag{4}$$

$$\sum_{\substack{u\in Outer_a\\ v\in Outer_a}} \Delta_G[T](u,v)^p = \sum_{\substack{u\in Outer_b\\ v\in Outer_b}} \Delta_G[T](u,v)^p = 0, \text{ and} \tag{5}$$

$$\sum_{\substack{u\in Outer_a\\ v\in Outer_b}} \Delta_G[T](u,v)^p = (n^{3p})^2\cdot k^p. \tag{6}$$

Note that Eq. (4) also holds true for $Outer_b$ and hence, using Eqs. (3)–(6) in (2),

$$\|D_T - D_G\|_{F,p}^p \le n^{2+p} + 4n^{1+4p} + 2n^{6p}k^p < 5n^{1+4p} + 2n^{6p}k^p \le n^{6p}(1+2k^p). \tag{7}$$

On the other hand, Eq. (2) yields

$$\begin{aligned}\|D_T - D_G\|_{F,p}^p \ &\ge\ 1 + \sum_{\substack{u\in Outer_a\\ v\in Outer_a}} \Delta_G[T](u,v)^p + \sum_{\substack{u\in Outer_b\\ v\in Outer_b}} \Delta_G[T](u,v)^p + 2\cdot \sum_{\substack{u\in Outer_a\\ v\in Outer_b}} \Delta_G[T](u,v)^p\\ &>\ 2n^{6p}\cdot k^p.\end{aligned} \tag{8}$$

In order to prove the inapproximability of the fixed edges version of DAST with respect to $\|\cdot\|_{L,p}$, assume that there exists a constant-factor approximation algorithm $\mathcal{A}$ that computes a tree $T_{\mathcal{A}}$. Then, using Eqs. (7) and (8), we obtain

$$\begin{aligned}0\ &\le c^p\cdot \|D_{T_{\text{opt}}} - D_G\|_{L,p}^p - \|D_{T_{\mathcal{A}}} - D_G\|_{L,p}^p\\ &\le c^p\cdot n^{6p}\cdot (1+2\Delta_G[T_{\text{opt}}](a,b)^p) - 2n^{6p}\cdot \Delta_G[T_{\mathcal{A}}](a,b)^p\\ &\le 2n^{6p}\cdot c^p\cdot \Big[\Big(\frac{1}{2} + \Delta_G[T_{\text{opt}}](a,b)^p\Big) - \Delta_G[T_{\mathcal{A}}](a,b)^p\Big].\end{aligned} \tag{9}$$

Therefore, again assuming $\Delta_G[T_{\text{opt}}](a,b) \ge 1$,

$$\Delta_G[T,\mathcal{A}](a,b)\ \le \delta\cdot \sqrt[p]{\frac{1}{2} + \Delta_G[T_{\text{opt}}](a,b)^p} \le 2\delta\cdot \Delta_G[T_{\text{opt}}](a,b). \tag{10}$$

Hence, applying algorithm $\mathcal{A}$ on the above construction, we can compute a spanning tree T' of G' with $E_0 \subseteq E(T)$ such that $\|D_{T'} - D_{G'}\|_{L,p} \le \delta\cdot \|D_{T_{\text{opt}}} - D_{G'}\|_{L,p}$. By Lemma 15, this implies P = NP. □

Theorem 18. *Unless* P = NP, *there is no polynomial-time constant-factor approximation algorithm for the fixed edges version of* DAST *with respect to* $\|\cdot\|_1$.

Proof. Let $(\mathcal{C}, \mathcal{S}, k)$ be a 2HS instance and let $G' = G(\mathcal{C}, \mathcal{S}, k, j)$ be a refined graph representation of $(\mathcal{C}, \mathcal{S}, k)$ for some $j \in \mathbb{N}_+$. Let n denote the number of vertices in G'. Define $\ell \overset{\text{def}}{=} 2n^2$. We construct a graph G consisting of a chain $G'_1, \ldots, G'_{\ell^2}$ of ℓ^2 copies of G', where the vertex b (denoted b_i in G) of G'_i is connected to the vertex a of G'_{i+1} (denoted a_{i+1} in G) by the edge $\{b_i, a_{i+1}\}$ in G. Additionally, G consists of

- four vertices c, c', d', and d connected by edges $\{c, c'\}$ and $\{d, d'\}$,
- two edges $\{c', a_1\}$ and $\{b_{\ell^2}, d'\}$, and
- two paths p' and q, each of length $\ell + 1$, connecting c with c' and d with d', respectively.

(Note how this construction is—except for the path's length and number of copies of G'—very similar to the one used in the proof of Lemma 16. Therefore, Fig. 6 may serve as an illustration.)

Let E'_i be the set of edges in all clause paths in G'_i. Define the set E_0 of fixed edges as

$$E_0 \overset{\text{def}}{=} \bigcup_{i=1}^{\ell^2} E'_i\ \cup\ E(p)\ \cup\ E(q).$$

Then, the solution tree T to the fixed-edge version of DAST with respect to $\|\cdot\|_1$ on G must contain p and q completely. However, it must not include $\{c, c'\}$ and $\{d, d'\}$, since otherwise there would be a cycle in the spanning tree. Hence, $\Delta_G[T](d, d') = \Delta_G[T](c, c') = \ell$. For any spanning tree T and any vertex $v \in V(G)$, define

$$\text{ColSum}(v) \overset{\text{def}}{=} \sum_{w\in V(G)} \Delta_G[T](v,w). \tag{11}$$

Choose a vertex x satisfying $\|D_T - D_G\|_1 = \text{ColSum}(x)$.

First, we will show that x is either c or d (which are handled analogously). Assume x is a vertex on p other than c, e.g., with distance i to c, then for any vertex u not on p, $\Delta_G[T](x,u) = \Delta_G[T](c,u) - i$, whereas the sum of the differences in distance to vertices on p is smaller by at least p (the difference in distance between c and c'—this can be seen by a simple coupling argument because we always have Δ's ranging from 1 to the maximum). Now, assume x is a vertex not on p or q, i.e., a vertex of G'_i. Without loss of generality, assume that $i < \frac{\ell^2}{2}$. For $j < i$, we pair each vertex $u \in G'_j$ with a vertex $v \in G'_{i+j}$. Then we find that

$$\begin{aligned}
&\Delta_G[T](c,u) + \Delta_G[T](c,v) - \Delta_G[T](x,u) - \Delta_G[T](x,v)\\
&\quad \geq \Delta_G[T](c,a_j) + \Delta_G[T](a_j,u) + \Delta_G[T](c,b_i) + \Delta_G[T](b_i,v)\\
&\qquad - \Delta_G[T](x,b_j) - \Delta_G[T](b_j,u) - \Delta_G[T](x,b_i) - \Delta_G[T](b_i,v)\\
&\quad = (\Delta_G[T](c,a_j) - \Delta_G[T](x,b_i)) + (\Delta_G[T](a_j,u) - \Delta_G[T](b_j,u)) + (\Delta_G[T](c,b_i) - \Delta_G[T](x,b_j))\\
&\quad \geq (\ell - n) + (0 - n) + (\ell - n) = 2\ell - 3n. \qquad (12)
\end{aligned}$$

For each remaining vertex $v \in G'_j$ for $j > 2(i-1)$, we find that

$$\Delta_G[T](c,v) - \Delta_G[T](x,v) = \Delta_G[T](c,b_i) + \Delta_G[T](b_i,v) - \Delta_G[T](x,b_i) - \Delta_G[T](b_i,v) \geq \ell - n.$$

The vertices on p and q are coupled again to find that for u on p and v on q we have

$$\begin{aligned}
&\Delta_G[T](c,u) + \Delta_G[T](c,v) - \Delta_G[T](x,u) - \Delta_G[T](x,v)\\
&\quad \geq \Delta_G[T](c,u) + \Delta_G[T](c,u) + \Delta_G[T](u,a_i) + \Delta_G[T](a_i,b_i) + \Delta_G[T](b_i,v)\\
&\qquad - \Delta_G[T](x,a_i) - \Delta_G[T](a_i,u) - \Delta_G[T](x,b_i) - \Delta_G[T](b_i,v)\\
&\quad \geq \Delta_G[T](c,u) + \Delta_G[T](c,u) + \Delta_G[T](a_i,b_i) - \Delta_G[T](x,a_i) - \Delta_G[T](x,b_i)\\
&\quad \geq -2n.
\end{aligned}$$

For any vertex v of G'_i we obtain $\Delta_G[T](c,v) - \Delta_G[T](x,v) \geq \ell - n$. As a result, we find that (t being the number of paired vertices in copies of G' by equation (12))

$$\begin{aligned}
\sum_{v \in V(G)} \Delta_G[T](c,v) - \Delta_G[T](x,v) &\geq t \cdot (2\ell - 3n) + (\ell^2 n - t + n)(\ell - n) - \frac{\ell}{2}n - 2\ell n\\
&\geq t \cdot (2n^2 - 2n) + 4n^3(n^4 - 1) > 0.
\end{aligned}$$

Therefore, either c or d will have the largest column sum of any vertex in G.

Assume that there exists a polynomial-time constant-factor approximation algorithm $\mathcal{A}'$ for the fixed-edge version of DAST with respect to $\|\cdot\|$. That is, for some $\delta > 0$, the algorithm outputs, on G, a spanning tree $T_{\mathcal{A}'}$ of G such that

$$\|D_{T_{\mathcal{A}'}} - D_G\|_1 \leq \delta \cdot \|D_{T_{\text{opt}}} - D_G\|_1. \qquad (13)$$

We know that the maximal column-sum is taken at c or d. Without loss of generality, $\|D_T - D_G\|_1 = \text{ColSum}(c)$. We show how to obtain an algorithm $\mathcal{A}$ from $\mathcal{A}'$ which constructs a spanning tree $T_{\mathcal{A}}$ for G satisfying Eq. (13) and having the property that for any i and h we have $\Delta_{G'_i}[T_{\mathcal{A}}](a_i,b_i) = \Delta_{G'_h}[T_{\mathcal{A}}](a_h,b_h)$. The algorithm works as follows. First, run $\mathcal{A}'$ on G, producing a spanning tree $T_{\mathcal{A}'}$ for G that satisfies Eq. (13). Then, select a G'_i for which $\Delta_{G'_i}[T_{A'}](a_i,b_i)$ is minimal and replace—for every G'_h for which $\Delta_{G'_h}[T_{\mathcal{A}'}](a_h,b_h)$ is *not* minimal—the respective part of $T_{\mathcal{A}'}$ by the spanning tree constructed for G'_i. Replacing the spanning tree increases the column-sum of c by at most n^2 (the Δ's in the graph whose spanning tree was replaced). However, it also decreases the column sum of c by at least ℓ (the Δ's to the vertices in q); thus, Eq. (13) still holds. Hence, we can replace the individual spanning tree for the G'_i until for all G'_i, $\Delta_{G'_i}[T](a,b)$ is identical. We may thus assume for the rest of the proof that in any spanning tree T for G, all G'_i have spanning trees with the same $\Delta_G[T](a,b)$. Define $k \overset{\text{def}}{=} \Delta_G[T](a,b)$.

Therefore, for vertex c, we have

$$\begin{aligned}
\text{ColSum}(c) &= \sum_{w \in V(p)} \Delta_G[T](c,w) + \sum_{i=;\ell^2} \sum_{w \in V(G'_i)} \Delta_G[T](c,w) + \sum_{w \in V(q)} \Delta_G[T](c,w)\\
&\geq \frac{1}{4}\ell^2 + n \cdot \sum_{i=1}^{\ell^2} (\ell + (i-1) \cdot k) + (\ell + 1) \cdot (\ell + \ell^2 \cdot k) + \frac{1}{4}\ell^2\\
&= \frac{1}{2}\ell^2 + n \cdot \left(\ell^3 + \frac{1}{2}\ell^4 \cdot k - \frac{1}{2}\ell^2 \cdot k\right) + \ell^2 + \ell + \ell^3 \cdot k + \ell^2 \cdot k\\
&= \frac{3}{2}\ell^2 + \ell^3 \cdot n + \ell + k \cdot \left(\frac{1}{2}\ell^4 \cdot n - \frac{1}{2}\ell^2 \cdot n + \ell^3 + \ell^2\right). \qquad (14)
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\mathrm{ColSum}(c) &= \sum_{w\in V(p)} \Delta_G[T](c,w) + \sum_{i=1}^{\ell^2} \sum_{w\in V(G'_i)} \Delta_G[T](c,w) + \sum_{w\in V(q)} \Delta_G[T](c,w) \\
&\le \frac{1}{4}\ell^2 + n\cdot \sum_{j=1}^{\ell^2} (\ell + (j-1)\cdot k + n) + (\ell+1)\cdot(\ell + \ell^2\cdot k) + \frac{1}{4}\ell^2 \\
&= \frac{1}{2}\ell^2 + n\cdot\left(\ell^3 + \frac{1}{2}\ell^4\cdot k - \frac{1}{2}\ell^2\cdot k + \ell^2\cdot n\right) + \ell^2 + \ell + \ell^3\cdot k + \ell^2\cdot k \\
&= \frac{3}{2}\ell^2 + \ell^3\cdot(1+n) + \ell + k\cdot\left(\frac{1}{2}\ell^4\cdot n - \frac{1}{2}\ell^2\cdot n + \ell^3 + \ell^2\right). \qquad (15)
\end{aligned}$$

Using Eqs. (14) and (15), we conclude from Eq. (13) that for $n \ge 2$, we have

$$\begin{aligned}
0 \ge \|D_{T_{\mathcal{A}}} - D_G\|_1 - \delta\cdot\|D_{T_{\mathrm{opt}}} - D_G\|_1 &\ge \frac{3}{2}\ell^2 + \ell^3\cdot n + \ell + k_{\mathcal{A}}\cdot\left(\frac{1}{2}\ell^4\cdot n - \frac{1}{2}\ell^2\cdot n + \ell^3 + \ell^2\right) \\
&\quad - \delta\cdot\left(\frac{3}{2}\ell^2 + \ell^3\cdot(n+1) + \ell + k_{\mathrm{opt}}\cdot\left(\frac{1}{2}\ell^4\cdot n - \frac{1}{2}\ell^2\cdot n + \ell^3 + \ell^2\right)\right) \\
&\ge \frac{\frac{3}{2}\ell^2 + \ell^3\cdot n + \ell}{\frac{1}{2}\ell^4\cdot n - \frac{1}{2}\ell^2\cdot n + \ell^3 + \ell^2} + k_{\mathcal{A}} - \delta\cdot\left(\frac{\frac{3}{2}\ell^2 + \ell^3\cdot(n+1) + \ell}{\frac{1}{2}\ell^4\cdot n - \frac{1}{2}\ell^2\cdot n + \ell^3 + \ell^2} + k_{\mathrm{opt}}\right) \\
&\ge k_{\mathcal{A}} - \delta\cdot(1 + k_{\mathrm{opt}}).
\end{aligned}$$

Assuming that $k_{\mathrm{opt}} \ge 1$ we thus find $k_{\mathcal{A}} \le 2\delta\cdot k_{\mathrm{opt}}$. By Lemma 15, this implies P = NP. □

5. 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 [2,23]:

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

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

Problem: CAST (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 $\|c_G - c_T\|_r \le \gamma$?

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

Theorem 19. CAST *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: in our new graph representation $G^*_{a,b}(\mathcal{C}, L) = (V^*, E^*)$ we omit the hull edges, i.e., we have

$$V^* = V$$
$$E^* = E \setminus \left\{\{l_\mu, l_\nu\} \mid \mu,\nu \in \{1,\dots,3m\} \text{ and } \mu \ne \nu\right\}$$

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 $\|c_{G^*_{a,b}(\mathcal{C},L)} - c_T\|_1 \le \gamma$. In the following we may restrict ourselves to the cases where $m \ge 5$.

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:

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

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 $\|c_{G^*_{a,b}(\mathcal{C},L)} - c_T\|_1 \le \gamma$. 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

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

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

  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 $y \ne -x$, the total centrality of T can be estimated as

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

  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

$$c_1 = s + 3(2+3b)m$$
$$c_2 = s + 3(2+3b)m + 2$$
$$c_3 = 2s + 3(3+4b)m - 6(b+1) - 1$$
$$c_4 = 2s + 3(3+4b)m - 12(b+1) - 1$$
$$A = 6s(b+1) + 6m(s-1)(b+1) + 18m(m-1)(b+1)^2,$$

  we see that $0 \le c_1 \le c_3$ and $0 \le c_2 \le c_4$ for $m \ge 5$. 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 the central edges 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 $c_T(v)^{-1} \ge \hat{c}(v)^{-1} - 6(b+1)$.
  - For $v \in L$ it suffices to have $c_T(v)^{-1} \ge \hat{c}(v)^{-1}$.
  - For $v \in K$ we obtain $c_T(v)^{-1} \ge \hat{c}(v)^{-1} + 2(b+1)(4 - \deg_T(u))$ 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 $c_T(v)^{-1} \ge \hat{c}(v)^{-1} + 2(b+1)$. 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:

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

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. □

Remark 20. Observe that the graph representation used in the proof of Theorem 19 always produces planar graphs if X3C instances are assumed not to contain two or more identical clauses. That is, CAST with respect to $\|\cdot\|_{L,1}$ is NP-complete even when restricted to planar graphs.

6. 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. As an interesting technical 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, maybe we can expect polynomial-time solvability there.

Acknowledgements

The third author was supported by DFG grant Ma 870/5-1 (Leibnizpreis Ernst W. Mayr). The fourth author was 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). The fifth author was supported by Deutsche Telekom Stiftung and Studienstiftung des deutschen Volkes.

References

[1] B. Awerbuch, Complexity of network synchronization, J. ACM 32 (4) (1985) 804–823.
[2] M.A. Beauchamp, An improved index of centrality, Behav. Sci. 10 (1965) 161–163.
[3] U. Brandes, T. Erlebach (Eds.), Network Analysis: Methodological Foundations, in: LNCS, vol. 3418, Springer-Verlag, 2005.
[4] U. Brandes, D. Handke, NP-completeness results for minimum planar spanners, Discrete Math. Theor. Comp. Sci. 3 (1) (1998) 1–10.
[5] L. Cai, NP-completeness of minimum spanner problems, Discrete Appl. Math. 48 (2) (1994) 187–194.
[6] P.M. Camerini, G. Galbiati, F. Maffioli, Complexity of spanning tree problems: Part I, European J. Oper. Res. 5 (5) (1980) 346–352.
[7] L.P. Chew, There are planar graphs almost as good as the complete graph, J. Comput. System. Sci. 39 (2) (1989) 205–219.
[8] E. Dahlhaus, P. Dankelmann, W. Goddard, H.C. Swart, MAD trees and distance-hereditary graphs, Discrete Appl. Math. 131 (1) (2003) 151–167.
[9] M. Elkin, D. Peleg, $(1+\epsilon, \beta)$-spanner constructions for general graphs, SIAM J. Comput. 33 (3) (2004) 608–631.
[10] S.P. Fekete, J. Kremer, Tree spanners in planar graphs, Discrete Appl. Math. 108 (1–2) (2001) 85–103.
[11] R. Hassin, A. Tamir, On the minimum diameter spanning tree problem, Inform. Process. Lett. 53 (2) (1995) 109–111.
[12] J. Håstad, Some optimal inapproximability results, J. ACM 48 (4) (2001) 798–859.
[13] D.S. Johnson, J.K. Lenstra, A.H.G. Rinnooy Kan, The complexity of the network design problem, Networks 8 (1978) 279–285.
[14] O. Kariv, S.L. Hakimi, An algorithmic approach to network location problems. I: The p-centers, SIAM J. Appl. Math. 37 (3) (1979) 513–538.
[15] D.-H. Kim, J.D. Noh, 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) (2001) 432–450.
[17] D. Kratsch, H.-O. Le, H. Müller, E. Prisner, D. Wagner, Additive tree spanners, SIAM J. Discrete Math. 17 (2) (2003) 332–340.
[18] A.L. Liestman, T.C. Shermer, Additive graph spanners, Networks 23 (4) (1993) 343–363.
[19] D. Peleg, E. Reshef, Low complexity variants of the arrow distributed directory, J. Comput. System. Sci. 63 (3) (2001) 474–485.
[20] D. Peleg, A.A. Schäffer, Graph spanners, J. Graph Theory 13 (1) (1989) 99–116.
[21] D. Peleg, J.D. Ullman, An optimal synchronizer for the hypercube, SIAM J. Comput. 18 (4) (1989) 740–747.
[22] E. Prisner, Distance approximating spanning trees, in: Proc. STACS'97, in: LNCS, vol. 1200, Springer-Verlag, 1997, pp. 499–510.
[23] G. Sabidussi, The centrality index of a graph, Psychometrica 31 (1966) 581–603.