

Almost Tight Approximation Hardness for Single-Source Directed k -Edge-Connectivity

Chao Liao ✉

Shanghai Jiao Tong University, China

Qingyun Chen ✉

University of California, Merced, CA, USA

Bundit Laekhanukit ✉

Shanghai University of Finance and Economics, China

Yuhao Zhang ✉

Shanghai Jiao Tong University, China

Abstract

In the k -outconnected directed Steiner tree problem (k -DST), we are given an n -vertex directed graph $G = (V, E)$ with edge costs, a connectivity requirement k , a root $r \in V$ and a set of terminals $T \subseteq V$. The goal is to find a minimum-cost subgraph $H \subseteq G$ that has k edge-disjoint paths from the root vertex r to every terminal $t \in T$. The problem is NP-hard, and inapproximability results are known in several parameters, e.g., hardness in terms of n : $\log^{2-\varepsilon} n$ -hardness for $k = 1$ [Halperin and Krauthgamer, STOC'03], $2^{\log^{1-\varepsilon} n}$ -hardness for general case [Cheriyān, Laekhanukit, Naves and Vetta, SODA'12], hardness in terms of k [Cheriyān et al., SODA'12; Laekhanukit, SODA'14; Manurangsi, IPL'19] and hardness in terms of $|T|$ [Laekhanukit, SODA'14].

In this paper, we show the approximation hardness of k -DST for various parameters.

- $\Omega(|T|/\log |T|)$ -approximation hardness, which holds under the standard complexity assumption $\text{NP} \neq \text{ZPP}$. The inapproximability ratio is tightened to $\Omega(|T|)$ under the Strongish Planted Clique Hypothesis [Manurangsi, Rubinfeld and Schramm, ITCS 2021]. The latter hardness result matches the approximation ratio of $|T|$ obtained by a trivial approximation algorithm, thus closing the long-standing open problem.
- $\Omega(2^{k/2}/k)$ -approximation hardness for the general case of k -DST under the assumption $\text{NP} \neq \text{ZPP}$. This is the first hardness result known for survivable network design problems with an inapproximability ratio exponential in k .
- $\Omega((k/L)^{L/4})$ -approximation hardness for k -DST on L -layered graphs for $L \leq O(\log n)$. This almost matches the approximation ratio of $O(k^{L-1} \cdot L \cdot \log |T|)$ achieved in $O(n^L)$ -time due to Laekhanukit [ICALP'16].

We further extend our hardness results in terms of $|T|$ to the undirected cases of k -DST, namely the single-source k -vertex-connected Steiner tree and the k -edge-connected group Steiner tree problems. Thus, we obtain $\Omega(|T|/\log |T|)$ and $\Omega(|T|)$ approximation hardness for both problems under the assumption $\text{NP} \neq \text{ZPP}$ and the Strongish Planted Clique Hypothesis, respectively. This again matches the upper bound obtained by trivial algorithms.

2012 ACM Subject Classification Mathematics of computing $\rightarrow$ Approximation algorithms; Mathematics of computing $\rightarrow$ Paths and connectivity problems

Keywords and phrases Directed Steiner Tree, Hardness of Approximation, Fault-Tolerant and Survivable Network Design

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.89

Category Track A: Algorithms, Complexity and Games

Related Version Full Version: <https://arxiv.org/abs/2202.13088>

Funding Qingyun Chen: Qingyun Chen is supported in part by NSF grants CCF-2121745 and CCF-1844939.



© Chao Liao, Qingyun Chen, Bundit Laekhanukit, and Yuhao Zhang;
licensed under Creative Commons License CC-BY 4.0

49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).

Editors: Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff;

Article No. 89; pp. 89:1–89:17



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Bundit Laekhanukit: Bundit Laekhanukit is supported by the 1000-talent plan award by the Chinese government, supported by Science and Technology Innovation 2030 — “New Generation of Artificial Intelligence” Major Project No.(2018AAA0100903), NSFC grant 61932002, Program for Innovative Research Team of Shanghai University of Finance and Economics (IRTSHUFE) and the Fundamental Research Funds for the Central Universities.

Yuhao Zhang: Yuhao Zhang is supported by National Natural Science Foundation of China (NSFC) under Grant No. 62102251.

1 Introduction

Fault-Tolerant and Survivable Network Design have been an active area of research for decades as enterprises depend more on communication networks and distributed computing. The need to design a network that can operate without disruption when one or more components fail has been growing dramatically. Henceforth, network scientists have formulated many models to address these problems. Amongst them, the simplest and arguably most fundamental problem in the area is the *minimum-cost k -outconnected spanning subgraph* (k -OCSS) problem that captures the problem of designing a multi-casting network with survivability property. The k -OCSS problem is a generalization of the *minimum spanning tree* and the *minimum-cost arborescence* problems, where the goal is to design a network that can operate under failures of at most $k - 1$ points. More formally, k -OCSS asks to find a minimum-cost subgraph such that the root vertex is k -connected to every other vertex.

In this paper, we study the analog of k -OCSS in the presence of Steiner vertices, namely the *k -outconnected directed Steiner tree* problem (k -DST): Given a directed graph $G = (V, E)$ with cost c_e on arcs, a root vertex r and a set of terminals T , the goal is to find a minimum-cost subgraph $H \subseteq G$ such that H has k edge-disjoint paths from the root r to every terminal $t \in T$, i.e., the root remains connected to every terminal even after the removal of $k - 1$ arcs. The k -DST problem is a natural generalization of the classical *directed Steiner tree* problem (DST) to high connectivity settings.

The undirected counterpart of k -DST is the *minimum-cost single source k -edge-connected Steiner tree* problem, which admits a factor-two approximation algorithm [28], and the vertex-connectivity variant admits an $O(k \log k)$ -approximation algorithm due to Nutov [36]. The k -DST problem, on the other hand, has no non-trivial approximation algorithm for $k \geq 3$, except for the special case of L -layered graph, which admits $O(k^L \cdot L \cdot \log |T|)$ -approximation algorithm due to Laekhanukit [33]. The cases of $k = 1$ and $k = 2$ are also notorious problems themselves, as both admit polylogarithmic approximation algorithms that run in quasi-polynomial time, but no polynomial-time approximation algorithms with sub-polynomial approximation. It has been long-standing open problems whether such algorithms exist for DST and 2-DST.

In this paper, we obtain several inapproximability results for k -DST. First, we show an approximation hardness of $\Omega(|T|/\log |T|)$ for k -DST under $\text{NP} \neq \text{ZPP}$, which holds when k is larger than $|T|$, thus implying that a trivial $|T|$ -approximation algorithm for the problem is tight up to the lower order term.

► **Theorem 1.** *For $k > |T|$, unless $\text{NP} = \text{ZPP}$, it is hard to approximate the k -DST problem to within a factor of $\Omega(|T|/\log |T|)$.*

Assuming the *Strongish Planted Clique Hypothesis* (SPCH) [35], our hardness result is tight up to a constant factor, and it, indeed, rules out $f(|T|) \cdot \text{poly}(n)$ -time $o(|T|)$ -approximation algorithm for any function f depending only on $|T|$. See discussion in the appendices of the full version of this paper.

■ **Table 1** Summary of the results for k -DST.

Parameter	Lower Bound (This paper)	Lower Bound (Previous)	Upper Bound
Connectivity k	$\Omega(2^{k/2}/k)$ Theorem 4	$\Omega(k/\log k)$ [34]	unknown for general $k \geq 3$
Connectivity k , Depth L	$\Omega((k/L)^{(1-\epsilon)L/4-2})$ Theorem 3	$\Omega(k/\log k)$ [34]	$O(k^{L-1} \cdot L \cdot \log T)$ [33]
Terminals $ T $	$\Omega(T /\log T)$ Theorem 1	$ T ^{1/4-\epsilon}$ [32]	$ T $ folklore

► **Theorem 2.** *Assuming the Strongish Planted Clique Hypothesis, there is no $f(|T|) \cdot \text{poly}(n)$ -time $o(|T|)$ -approximation algorithm for the k -DST problem.*

Next, we show that the k -DST admits no $O((k/L)^{L/4})$ -approximation algorithm even on an L -layered graph, which consists of L parts, called *layers*, and every arc joins a vertex from the i -th layer to the $(i+1)$ -th layer.

► **Theorem 3.** *It is hard to approximate the k -DST problem on L -layered graphs $G = (V, E)$ for $\Omega(1) \leq L \leq O(\log |V|)$ to within a factor of $\Omega((k/L)^{(1-\epsilon)L/4-2})$ for any constant $\epsilon > 0$, unless $\text{NP} = \text{ZPP}$.*

In addition, we obtain an approximation hardness exponential in k by setting a different parameter in the reduction, which improves upon the previously known approximation hardness of $\Omega(k/\log k)$ due to Manurangsi [34] (which is in turn based on the two previous results [32, 11]), and is the first known approximation hardness for connectivity problems whose ratio is exponential in the connectivity requirement.

► **Theorem 4.** *For $k < |T|$, it is hard to approximate the k -DST problem to within a factor of $\Omega(2^{k/2}/k)$, unless $\text{NP} = \text{ZPP}$.*

Using the technique of Cheriyan, Laekhanukit, Naves and Vetta [11], which is based on the padding technique introduced by Kortsarz, Krauthgamer and Lee [31], we extend our hardness result to the undirected counterpart of k -DST, namely, the *single source k -vertex-connected Steiner tree* problem (k -ST) (a.k.a. *undirected rooted subset k -connectivity*, *shorty*, *rooted- k -VC*) and the special case of k -DST, namely *k -edge-connected group Steiner tree* problem (k -GST).

The latter problem is a natural fault-tolerant generalization of the classical group Steiner tree problem [19], which has been studied in [29, 24, 6, 3]. To the best of our knowledge, a non-trivial approximation algorithm for this problem is known only for $k = 1, 2$. For $k \geq 3$, only a bicriteria approximation algorithm, where the connectivity requirement can be dropped by a factor $O(\log n)$, is known in [6]. Nevertheless, a trivial $|T|$ -approximation algorithm exists for all values of k and we also show its tightness (up to the lower order term) for sufficiently large k .

► **Theorem 5.** *For $k > |T|$, unless $\text{NP} = \text{ZPP}$, it is hard to approximate the k -ST problem to within a factor of $\Omega(|T|/\log |T|)$.*

► **Theorem 6.** *For $k > |\mathcal{T}|$, unless $\text{NP} = \text{ZPP}$, it is hard to approximate the k -GST problem to within a factor of $\Omega(|\mathcal{T}|/\log |\mathcal{T}|)$, where $|\mathcal{T}|$ is the number of groups.*

Related work

The k -DST is well-studied in the special case where all vertices are terminals. This problem is, as mentioned, known as the k -outconnected spanning subgraph problem (k -OCSS), which admits polynomial-time algorithms due to the seminal work of Frank and Tardos [17] (also, see [16]). However, while k -OCSS is polynomial-time solvable, its undirected counterpart is NP-hard. Nevertheless, Frank-Tardos's algorithm has been used as subroutines to derive a 2-approximation algorithm for the undirected variant of k -DST and its generalization [30].

In the presence of Steiner vertices, k -DST becomes much harder to approximate. For the case of $k = 1$, the best known polynomial-time approximation algorithms are $|T|^\epsilon$, for any constant $\epsilon > 0$, due to the work of Charikar et al. [8], and the same approximation ratio (with an additional log factor) applies for the case $k = 2$ due to the work of Grandoni and Laekhanukit [22]. These two special cases of k -DST, especially for the case $k = 1$, have been perplexing researchers for many decades as it admits polylogarithmic approximation algorithms in quasi-polynomial-time, whereas there is no known sub-polynomial-approximation algorithm for the problems; see, e.g., [8, 23, 20, 22]. It has been a long-standing open problem whether polylogarithmic or even sub-polynomial-approximation ratios can be achieved in polynomial time. Some special cases of k -DST have been studied in the literature. Laekhanukit [33] studied the k -DST instances on L -layered graphs, and its extensions to the L -shallow instances, and presented an $O(k^L \cdot L \cdot \log |T|)$ -approximation algorithm that runs in $n^{O(L)}$ time. Polynomial-time polylogarithmic approximation algorithms for k -DST are known in quasi-bipartite graphs [7, 37] (also, see [18, 27] for the case $k = 1$, which matches the approximation lower bound of $(1 - \epsilon) \ln k$, assuming $P \neq NP$, inherited from the *Set Cover* problem [14, 12]).

For the undirected case of $k = 1$, namely the Steiner tree problem, it admits a 1.39-approximation algorithm due to the breakthrough result of Byrka et al. [2] and admits a $\frac{73}{60}$ -approximation algorithm on quasi-bipartite graphs due to the work of Goemans et al. [21]. For $k \geq 2$, the problems on undirected graphs are branched into edge and vertex connectivity variants. This is not the case for directed graphs as there is a simple approximation-preserving reduction from edge-connectivity to vertex-connectivity and vice versa. For the edge-connectivity problem, it admits a 2-approximation algorithm by using Frank-Tardos's algorithm as subroutines when there is no Steiner vertex [30], and a 2-approximation algorithm via *iterative rounding* due to the seminal result of Jain [28], which also applies for the more general case of the *edge-connectivity survivable network design* problem. For the vertex-connectivity problem, there is a 2-approximation algorithm for $k = 2$ due to Fleischer, Jain and Williamson [15], but the problem becomes hard polynomial in k , for sufficiently large k [11], assuming $P \neq NP$. The best known approximation algorithm for the *single-source k -vertex-connectivity* problem on undirected graphs is $O(k \log k)$ due to Nutov [36].

The network design problem where the connectivity requirements are between pairs of vertices is sometimes called point-to-point network design. A natural generalization is to extend the requirements to be between subsets of vertices, called groups. The classical problem in this genre is the well-studied *group Steiner tree* problem (see, e.g., [19, 9, 10, 25, 26, 4]). The group Steiner tree problem admits an approximation ratio of $O(\log q \log n)$ [19], which requires a probabilistic metric-tree embedding [1, 13]. This approximation ratio is almost tight as it matches the lower bound of $O(\log^{2-\epsilon} n)$, for any constant $\epsilon > 0$, due to the hardness result of Halperin and Krauthgamer [26] assuming $NP \not\subseteq ZPTIME(n^{\text{polylog}(n)})$; also, see the improved hardness result in [23]. The fault-tolerant variant of the group Steiner tree problem is called the k -edge-connected group Steiner tree problem, studied in [29, 24, 6, 3]. As mentioned, true approximation algorithms for this problem are known only for $k = 1, 2$ [19, 29, 24]. For $k \geq 3$, only a bi-criteria approximation algorithm is known [6].

Organization

Section 2 is devoted to preliminary notations, definitions and facts. The reductions for k -DST are presented in Section 3, 4 and 5. We give some intuitions on the techniques in Section 3 and describe the reduction for hardness in terms of $|T|$ in Section 4 and for hardness in terms of k in Section 5. In Section 6 we briefly discuss the results for k -DST. Finally, we extend our techniques to undirected graphs and tighten the $\Omega(|T|/\log |T|)$ lower bound to $\Omega(|T|)$ under a stronger complexity hypothesis. These results are presented in the appendices of the full version of this paper.

2 Preliminaries

We use a standard graph terminology. Let $G = (V, E)$ be any graph, which can be either directed or undirected. For undirected graphs, we refer to the elements in E as the “edges” of G and denote by $\deg_G(v)$ the number of edges incident to a vertex $v \in V$. For directed graphs, we refer to the elements in E as the “arcs” of G and denote by $\text{indeg}_G(v)$ the number of arcs entering v . The notation for an edge/arc is (u, v) , or sometimes $u \rightarrow v$ for an arc. For a path between vertex u and w , we call it a (u, w) -path and write it as $(u, v, \dots, w)$ for both directed and undirected graphs, or $u \rightarrow v \rightarrow \dots \rightarrow w$ for only directed graphs. The graphs may have multiple edges/arcs between two same vertices u and v , and both $\deg_G(v)$ and $\text{indeg}_G(v)$ count multiple ones. We drop G from the notations when it is clear from the context. When more than one graph is considered, we use $V(G)$ to clarify the vertex set of G , and $E(G)$ the edge/arc set.

k -(Edge)-Connected Directed Steiner Tree

The k -(edge)-connected directed Steiner tree problem (k -DST) is defined as follows. An input instance is of the form (k, G, r, T) where $k \in \mathbb{Z}_{\geq 1}$ is the connectivity requirement, $G = (V, E)$ is a directed graph with weight (or cost) on arcs $c : E \rightarrow \mathbb{Q}_{\geq 0}$, $r \in V$ is called root and $T \subseteq V$ is a set of terminals. A subgraph $H = (V, F)$ of G is k -connected if there exist k edge-disjoint paths in H from r to t for each terminal $t \in T$. Sometimes we also refer to a k -connected subgraph as a feasible solution to the k -DST problem. The problem is to find a k -connected subgraph $H = (V, F)$ of minimum cost $c(H) = \sum_{e \in F} c(e)$.

Here, we define the problem in terms of edge-connectivity. A vertex-connectivity variant is defined similarly except that it asks for (openly) vertex-disjoint paths instead of edge-disjoint paths. Both variants are equivalent in terms of approximability on directed graphs because there exist straightforward polynomial-time approximation-preserving reductions from any one to the other.

(Minimum) Label Cover

An instance of the label cover problem is given by an (undirected) bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$, a set of labels $\Sigma = [g]$ and (projection) constraints $\pi_{uv} : \Sigma \rightarrow \Sigma$ on each edge $(u, v) \in \mathcal{E}$. A multilabeling $\sigma : \mathcal{U} \cup \mathcal{V} \rightarrow 2^\Sigma$ is a subset of labels assigned to each vertex. We say that σ covers an edge $(u, v) \in \mathcal{E}$ if $\pi_{uv}(a) = b$ for some $a \in \sigma(u)$ and $b \in \sigma(v)$. A multilabeling is *feasible* if it covers all the edges of $\mathcal{E}$. The problem asks for a feasible multilabeling σ with minimum cost $c(\sigma) = \sum_{u \in \mathcal{U} \cup \mathcal{V}} |\sigma(u)|$.

Manurangsi [34] proved that the label cover problem has a hardness gap in terms of the maximum degree of $\mathcal{G}$.

► **Theorem 7** ([34]). *For every positive integer $g > 1$, unless $\text{NP} = \text{ZPP}$, it is hard to approximate a label cover instance of maximum degree $O(g \log g)$ and alphabet size $O(g^4 \log^2 g)$ within a factor of g .*

The following corollary can be deduced straightforwardly.

► **Corollary 8.** *It is hard to approximate a label cover instance of maximum degree Δ and alphabet size $O(\Delta^4 / \text{polylog}(\Delta))$ to within a factor of $\Omega(\Delta / \log \Delta)$, or to within a factor of $\Omega(\Delta^{1-\varepsilon})$ for any constant $\varepsilon > 0$, unless $\text{NP} = \text{ZPP}$.*

To obtain the hardness results on k -DST (and related problems), we present reductions from the label cover problem on an instance $(\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E}), \Sigma = [g], \pi = \{\pi_{uv} : \Sigma \rightarrow \Sigma\}_{(u,v) \in \mathcal{E}})$ of maximum degree Δ . For the ease of presentation, let $\mathcal{U} = \{u_1, u_2, \dots\}$ and $\mathcal{V} = \{v_1, v_2, \dots\}$.

Finally, we prepare a technical lemma for future reference. We say that a subgraph I of G is an *induced matching* if 1) I is a matching, i.e., each vertex in G is the endpoint of at most one edge in $E(I)$ and 2) I is an induced subgraph, i.e., all edges with two endpoints both in $V(I)$ are included in $E(I)$.

► **Lemma 9** (Folklore). *Let $G = (U, V, E)$ be a bipartite graph of maximum degree Δ . There exist a partition of the edges $E = E_1 \cup E_2 \cup \dots \cup E_\Delta$ such that each E_i is a matching of G , and a partition $E'_1, E'_2, \dots, E'_\delta$ of E for some $\delta \leq 2\Delta^2$ such that each E'_i is an induced matching. Furthermore, such partitions can be found in polynomial time.*

3 Overview of the Reductions

To give some intuitions on how our reductions work, we dedicate this section to providing an overview. We have two main reductions, which are tailored for inapproximability results in different parameters, say $|T|$ and k .

Both of the reductions inherit approximation hardness from the same source – the label cover problem, denoted by $(\mathcal{G}, \Sigma, \pi)$. We design reductions that have a one-to-one correspondence between a feasible solution to the label cover problem and that to the k -DST problem, i.e.,

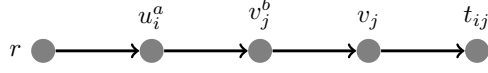
- **Completeness:** Given a feasible multilabeling σ of the label cover instance $(\mathcal{G}, \Sigma, \pi)$, there is a corresponding k -connected subgraph H of G such that $c(\sigma) = c(H)$.
- **Soundness:** Given a k -connected subgraph H of the k -DST instance, there is a corresponding feasible multilabeling σ of the label cover instance $(\mathcal{G}, \Sigma, \pi)$ such that $c(\sigma) = c(H)$.

We note that certain details in the ideas mentioned below are intentionally omitted, which can be found in later chapters.

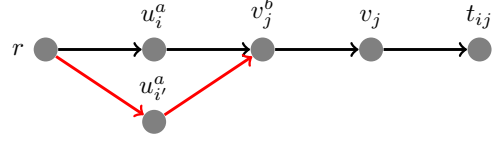
Basic Construction

First, we present the basic construction for the case $k = 1$, which is sufficient to maintain the completeness property (but not the soundness). We start by adding a root vertex r and terminal vertices t_{ij} , one for each edge $(u_i, v_j) \in \mathcal{E}$. Observe that each terminal corresponds to an edge in the label cover instance. Thus, we wish to make an (r, t_{ij}) -path in G to correspond to a labeling that covers an edge in $\mathcal{G}$, which we call a *cover path*. To be more precise, if the edge (u_i, v_j) is covered by a label pair (a, b) , then the corresponding *cover path* in G is the path $r \rightarrow u_i^a \rightarrow v_j^b \rightarrow v_j \rightarrow t_{ij}$ where the vertices u_i^a and v_j^b are simply added to the graph G . The appearance of the vertex $u_i^a \in V$ along the cover path can be interpreted as assigning the label a to the vertex $u_i \in \mathcal{U}$, and similarly, v_j^b means assigning the label b to

the vertex $v_j \in \mathcal{V}$. See Figure 1 for illustration. Note that any feasible multilabeling covers all the edges of $\mathcal{G}$, so we can collect all cover paths (and the involved vertices) to form a subgraph, where the root is already 1-connected to the terminals. By setting the weight of all arcs $r \rightarrow u_i^a$ and $v_j^b \rightarrow v_j$ to be 1 while leaving other arcs zero-cost, the subgraph has the same cost as the multilabeling.



■ **Figure 1** A cover path $(r, u_i^a, v_j^b, v_j, t_{ij})$.



■ **Figure 2** An illegal path $(r, u_{i'}^a, v_j^b, v_j, t_{ij})$.

However, the soundness property does not hold on the basic construction because it creates many *illegal* (r, t_{ij}) -paths. Such a path goes from the root to a terminal t_{ij} by using the route that differs from $(u_i, v_j) \in \mathcal{E}$, e.g., $r \rightarrow u_{i'}^a \rightarrow v_j^b \rightarrow v_j \rightarrow t_{ij}$, where $i' \neq u_i$; see Figure 2. This means that, although a solution is feasible to the 1-DST instance, the subgraph may not have a cover path for every terminal. Thus, the corresponding multilabeling may leave some edges in the label cover instance uncovered. To ensure that at least one cover path exists for every terminal, we need to modify our instance using the *padding arc* technique.

Padding Arcs

Consider an illegal (r, t_{ij}) -path in the basic construction. While we wish the (r, t_{ij}) -path to visit vertices u_i^a and v_j^b , which corresponds to a satisfying labeling to the edge (u_i, v_j) in the label cover instance, the illegal path instead visits $u_{i'}^a$ with $u_{i'} \neq u_i$. In particular, the illegal path exploits a cover path for some other edge $(u_{i'}, v_j)$ that share the same endpoint with (u_i, v_j) .

To prevent this from happening, we construct a zero-cost *padding path* from the root to the terminal t_{ij} that shares some arcs with the illegal path. These two paths are mutually exclusive in contributing to the edge-connectivity between the root r and the terminal t_{ij} . As we set the connectivity requirement to be the same as the indegree of t_{ij} , it forces all the padding paths for t_{ij} to be used in any feasible solution. Once all the padding paths are used to form $k - 1$ edge-disjoint (r, t_{ij}) , the only path available is forced to be a cover path.

Size of $|T|$ and k

The construction as mentioned above yields a one-to-one correspondence between the feasible solutions to k -DST and that of the label cover problem. However, the size of the construction in terms of the parameter k (resp., $|T|$) is too large comparing to the inapproximation factor of $\Omega(\Delta/\log \Delta)$ inherited from the label cover problem. Specifically, the value of k can be as large as the number of illegal paths, and the value of $|T|$ can be as large as $|\mathcal{E}| = |\mathcal{U}| \cdot \Delta$. Therefore, we need to optimize the size of $|T|$ and k , where we use two different techniques, one for each parameter.

- To control the size of the terminal set, we partition the edge set $\mathcal{E}$ into Δ **matchings**, and only **one** terminal is constructed for **each matching** rather than having one terminal for each edge. This, thus, reduces the size of T to Δ .¹

¹ Laekhanukit [32] applies a similar techniques using strong edge-coloring, which gives a worse factor of Δ^2 .

- To control the connectivity requirement k , we partition the edge set $\mathcal{E}$ into $\delta \leq 2\Delta^2$ **induced matchings** and create a d -ary tree structure with δ leaves, where d is an adjustable parameter. Roughly speaking, by exploiting the tree structure, we need to add to each terminal only d *padding arcs* for each of the $O(\log_d \delta)$ layers of the tree. Thus, the connectivity requirement k is reduced to $O(d \log_d \delta)$.

Generalized to Undirected Settings

We also apply the techniques mentioned above to the undirected settings. For the undirected k -connected Steiner tree problem (k -ST), we migrate the reduction for k -DST with a hardness result in terms of $|T|$ to its vertex-connectivity version, with necessary adaptations, thus reproducing the same $|T|/\log |T|$ inapproximability. The k -connected group Steiner tree problem (k -GST) is a generalization of the k -ST problem that turns out to be one of the key components in designing approximation algorithms for k -DST [38, 22] for $k = 1, 2$. By incorporating ideas from [5], we achieved the same $q/\log q$ inapproximability for even the edge-disjoint version of k -GST where q is the number of groups. The main difficulty is that undirected edges allow new illegal paths to enter a terminal. Fortunately, by equalizing the connectivity requirement and the size of a group, it turns out we can handle the extra paths. See the appendices in the full version of this paper for a complete presentation.

4 Inapproximability in Terms of the Number of Terminals

In this section, we discuss the hardness reduction that is tailored for the parameter $|T|$. Our reduction takes as input a label cover instance $(\mathcal{G}, \Sigma, \pi)$ and then produces a k -DST instance $(k, G = (V, E), r, T)$ as an output. The reduction runs in polynomial-time, and there is a one-to-one mapping between the solutions to the two problems. Thus, the inapproximability result of label cover is mapped to the inapproximability of k -DST directly. The main focus in this section is in reducing the number of terminals by exploiting edge-disjoint paths.

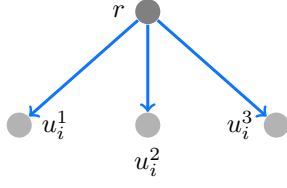
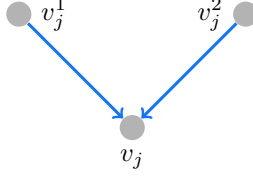
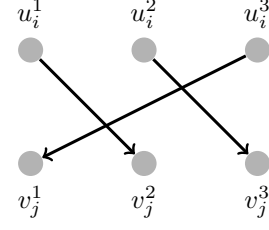
Base Construction

The construction starts from a basic building block.

1. First, create a graph G with a single vertex, the root r .
2. (Refer to Figure 3) For each $u_i \in \mathcal{U}$, create in G each vertex u_i^a from $A_i = \{u_i^a : a \in \Sigma\}$; connect r to u_i^a by an arc (r, u_i^a) of cost one.
3. (Refer to Figure 4) For each $v_j \in \mathcal{V}$, create a counterpart of it (also named v_j) in G and create each vertex v_j^b from $B_j = \{v_j^b : b \in \Sigma\}$; connect v_j^b to v_j by an arc (v_j^b, v_j) of cost one.
4. (Refer to Figure 5) For each $(u_i, v_j) \in \mathcal{E}$ and $a, b \in \Sigma$, connect u_i^a to v_j^b by a zero-cost arc if $\pi_{u_i v_j}(a) = b$.

Final Construction

Finally, we have to add some zero-cost arcs so called *padding arcs* to the base construction, which are meant to enforce the constraints of the label cover problem into the k -DST instance. We first partition the edge set $\mathcal{E}$ of the graph $\mathcal{G}$ in the label cover instance into Δ matchings, denoted by $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_\Delta$. This step can be done in polynomial-time due to Lemma 9. We then create a set of Δ terminals corresponding to these matchings.

■ **Figure 3** $\Sigma = \{1, 2, 3\}$.■ **Figure 4** $\Sigma = \{1, 2\}$.■ **Figure 5** $\pi_{u_i v_j}(x) = x \bmod 3 + 1$.

1. For each matching $\mathcal{E}_m$, add a terminal t_m to G and connect the counterpart of each $v_j \in V(\mathcal{E}_m) \cap \mathcal{V}$ in G to t_m . (Refer to Figure 6).

Next, we add some padding arcs to form padding paths that “kills” illegal paths.

2. Instead of connecting u_i^a to v_j^b for the edges $(u_i, v_j) \in \mathcal{E}$ such that $\pi_{u_i v_j}(a) = b$ directly, we add an internal node w_{ij}^{ab} and replace the original arc (u_i^a, v_j^b) in G by arcs (u_i^a, w_{ij}^{ab}) and (w_{ij}^{ab}, v_j^b) . (Here and thereafter refer to Figure 7.)
3. For each u_i^a , we connect r to u_i^a by $\deg_G(u_i)$ copies of an arc (r, u_i^a) .
4. For $1 \leq m \leq \Delta$, $(u_i, v_j) \in \mathcal{E}$ and $a, b \in \Sigma$ such that $\pi_{u_i v_j}(a) = b$, if $(u_i, v_j) \in \mathcal{E}_m$, then we add an arc (u_i^a, t_m) ; otherwise, we add an arc (w_{ij}^{ab}, t_m) . Thus, we finally have $|\Sigma| \cdot |\mathcal{E}_m|$ arcs from the vertex set $\{u_i^a : u_i \in \mathcal{U}, a \in \Sigma\}$ to t_m , and $|\Sigma| \cdot (|\mathcal{E}| - |\mathcal{E}_m|)$ arcs from the internal vertex set $\{w_{ij}^{ab} : \pi_{u_i v_j}(a) = b\}$ to t_m .
5. We set $k = \max_{1 \leq m \leq \Delta} \text{indeg}_G(t_m)$. To make the connectivity requirement uniform, we add $k - \text{indeg}_G(t_m)$ copies of an arc (r, t_m) for each terminal t_m .

Please see Figure 7 for an illustration. Observe that the connectivity requirement k is exactly the indegree of each terminal. Thus, all of its incoming arcs are needed in any feasible solution. Now, consider the edge $(u_i, v_{j'})$ in the figure, which is not in $\mathcal{E}_m$. It is possible that a feasible solution includes the path $r \rightarrow u_i^a \rightarrow w_{ij'}^{ab'} \rightarrow v_{j'}^{b'} \rightarrow t_m$, which is an illegal path for the terminal t_m . However, if we wish to route k -edge-disjoint paths between the root r and t_m , then the arc $w_{ij'}^{ab'} \rightarrow t_m$ must be used, and the only way to use this arc is to traverse from $u_i^a \rightarrow w_{ij'}^{ab'}$. This prevents the illegal path from using this arc, meaning that it cannot be included in any k -edge-disjoint (r, t_m) -paths. Less formally, we may say that it gets killed by the padding path $r \rightarrow u_i^a \rightarrow w_{ij'}^{ab'} \rightarrow t_m$.

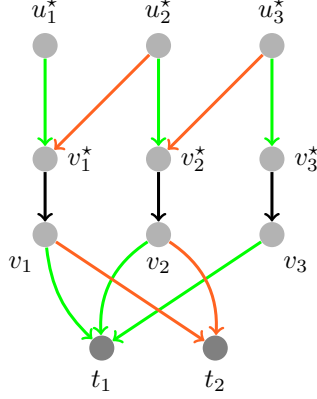
Why Do We Need A Matching?

Consider a terminal t_m . Our construction promises edge-disjoint cover paths from r to t_m for every edge in $\mathcal{E}_m$. However, if the edges in $\mathcal{E}_m$ do not form a matching, then two cover paths may share some edge. That is, the corresponding subgraph of a feasible multilabeling may have connectivity less than k , and the completeness property breaks.

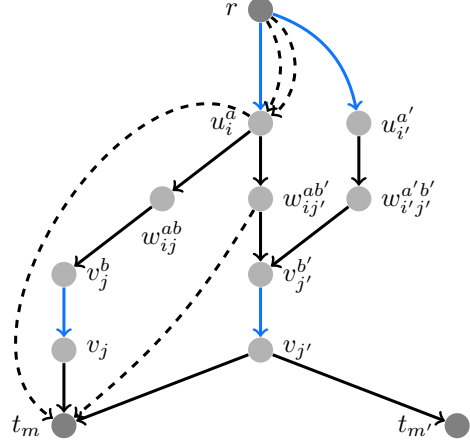
Next, we prove the one-to-one correspondence between solutions to the two instances. Wlog., assume that any solution to the k -DST problem contains all of the zero-cost arcs.

Completeness

Given a feasible multilabeling σ of the label cover instance $(\mathcal{G}, \Sigma, \pi)$, we show that there is a corresponding feasible subgraph $H = (V, F)$ of $G = (V, E)$ such that $c(\sigma) = c(H)$. The set F consists of three types of arcs: 1) all zero-cost arcs in G ; 2) the one-cost arcs (r, u_i^a) for each $u_i \in \mathcal{U}$ and $a \in \sigma(u_i)$; 3) the one-cost arcs (v_j^b, v_j) for each $v_j \in \mathcal{V}$ and $b \in \sigma(v_j)$. Clearly, $c(H) = c(\sigma)$ and the definition of F induces an injective mapping.



■ **Figure 6** The terminal t_1 corresponds to the matching $\{(1, 1), (2, 2), (3, 3)\}$ and t_2 corresponds to $\{(2, 1), (3, 2)\}$.



■ **Figure 7** Padding arcs are dashed. The matching $\mathcal{E}_m$ contains the edges (u_i, v_j) and $(u_{i'}, v_{j'})$, and the matching $\mathcal{E}_{m'}$ contains the edge $(u_i, v_{j'})$.

Next we prove the feasibility of H . That is, we will show that, for any terminal t_m , there exist k edge-disjoint paths from r to t_m . We will construct a set of such paths, namely P . Note that every arc entering t_m must be contained in a distinct path in P because $\text{indeg}_H(t_m) = k$. This gives four types of paths.

- (r, t_m) : The arc itself forms a path from r to t_m .
- (u_i^a, t_m) : We choose one of the zero-cost arcs (r, u_i^a) to combine with (u_i^a, t_m) to constitute a path $r \rightarrow u_i^a \rightarrow t_m$. Since $\mathcal{E}_m$ is a matching, by construction we know that the arc (u_i^a, t_m) has multiplicity one in G and thus the paths in this category are edge-disjoint. After selecting paths in this way, there are still $\deg_G(u_i) - 1$ unoccupied copies of the zero-cost arc (r, u_i^a) if $u_i \in V(\mathcal{E}_m)$; or $\deg_G(u_i)$ unoccupied copies otherwise.
- (w_{ij}^{ab}, t_m) : We choose one of the zero-cost arcs (r, u_i^a) and the arc (u_i^a, w_{ij}^{ab}) to constitute a path $r \rightarrow u_i^a \rightarrow w_{ij}^{ab} \rightarrow t_m$. If $u_i \in V(\mathcal{E}_m)$, such paths use $\deg_G(u_i) - 1$ copies of the zero-cost arc (r, u_i^a) , otherwise $\deg_G(u_i)$ copies are used. In both cases it is valid to do so because the first two categories of path leave enough copies. It is clear that the paths up to this point are edge-disjoint.
- (v_j, t_m) : In this case there is a unique $u_i \in \mathcal{U}$ such that $(u_i, v_j) \in \mathcal{E}_m$. It holds that $\pi_{u_i v_j}(a) = b$ for some $a \in \sigma(u_i), b \in \sigma(v_j)$ since σ covers the edge (u_i, v_j) . We choose the one-cost arcs $(r, u_i^a) \in F$ and $(v_j^b, v_j) \in F$ to constitute a path $r \rightarrow u_i^a \rightarrow w_{ij}^{ab} \rightarrow v_j^b \rightarrow v_j \rightarrow t_m$. Since $\mathcal{E}_m$ is a matching, the paths here are edge-disjoint. Note that previous paths only use arcs added in the final construction except for arcs of the form (u_i^a, w_{ij}^{ab}) , while here we only use arcs from the base construction. The construction of G guarantees (u_i^a, w_{ij}^{ab}) is not used by previous paths if $(u_i, v_j) \in \mathcal{E}_m$.

Therefore, we selected k edge-disjoint paths in H from r to t_m successfully.

Soundness

Given a k -connected subgraph $H = (V, F)$ (that contains all zero-cost arcs) of the k -DST instance (k, G, r, T) , we show that there is a corresponding feasible multilabeling σ of the label cover instance $(\mathcal{G}, \Sigma, \pi)$ such that $c(\sigma) = c(H)$. The multilabeling σ is specified by checking the one-cost arcs in H , i.e., set $\sigma(u_i)$ as $\{a \in \Sigma : \text{the one-cost arc } (r, u_i^a) \text{ is in } F\}$ for $u_i \in \mathcal{U}$ and set $\sigma(v_j)$ as $\{b \in \Sigma : (v_j^b, v_j) \in F\}$ for $v_j \in \mathcal{V}$. Clearly, $c(\sigma) = c(H)$ and the definition of σ induces an injective mapping.

We prove that there exist $a_i \in \sigma(u_i)$ and $b_j \in \sigma(v_j)$ such that $\pi_{u_i v_j}(a_i) = b_j$ for each edge $(u_i, v_j) \in \mathcal{E}$. Recall that $\mathcal{E}$ is partitioned into Δ matchings $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_\Delta$. Let us fix an arbitrary matching $\mathcal{E}_m$ and discuss the edges inside $\mathcal{E}_m$. Consider the set S of arcs in G coming into the terminal t_m . By our construction, S contains k arcs of the following categories:

1. $|\mathcal{E}_m|$ arcs of type (v_j, t_m) ;
2. $|\Sigma| \cdot (|\mathcal{E}| - |\mathcal{E}_m|)$ arcs of type $(w_{i'j'}^{a'b'}, t_m)$ (one for each $(u_{i'}, v_{j'}) \notin \mathcal{E}_m$ and $a', b' \in \Sigma$ such that $\pi_{u_{i'} v_{j'}}(a') = b'$);
3. $|\Sigma| \cdot |\mathcal{E}_m|$ arcs of type (u_i^a, t_m) (one for each $(u_i, v_j) \in \mathcal{E}_m$ and $a \in \Sigma$);
4. $k - |\mathcal{E}_m| - |\Sigma| \cdot |\mathcal{E}|$ arcs of type (r, t_m) .

Let P be the subgraph formed by the k edge-disjoint paths from r to t_m in H . The connectivity requirement forces that each arc in S must belong to some path in P , so we can also categorize the paths in P into the four types above. Let P_{2-4} be the subgraph consisting of all type-2,3,4 paths, and P_i similarly. We prove two observations:

▷ **Claim 10.** We have the following facts for paths.

- I: $\forall (u_{i'}, v_{j'}) \notin \mathcal{E}_m, \forall a', b' \in \Sigma$ such that $\pi_{u_{i'} v_{j'}}(a') = b', (u_{i'}^{a'}, w_{i'j'}^{a'b'}) \in P_2 \notin P_1$.
- II: $\forall (u_i, v_j) \in \mathcal{E}_m$ and $a, b \in \Sigma$ such that $\pi_{u_i v_j}(a) = b$, there are $\deg_G(u_i)$ arcs (r, u_i^a) in P_{2-4} .

Proof. Note that we need a type-2 path from all $w_{i'j'}^{a'b'}$ such that $\pi_{u_{i'} v_{j'}}(a') = b'$, and using the arc $(u_{i'}^{a'}, w_{i'j'}^{a'b'})$ is the only way to enter $w_{i'j'}^{a'b'}$ to form a type-2 path. So $(u_{i'}^{a'}, w_{i'j'}^{a'b'})$ belongs to P_2 and not in P_1 (edge disjoint with P_2).

For the second claim, there are $\deg_G(u_i) - 1$ edges $(u_i, v_j) \in \mathcal{E} \setminus \mathcal{E}_m$. Plugging in Claim-I implies that there are $\deg_G(u_i) - 1$ type-2 paths that use u_i^a . Moreover, because $(u_i, v_j) \in \mathcal{E}_m$, there is another type-3 path that use u_i^a . So, in total there are $\deg_G(u_i)$ paths in P_{2-4} that use u_i^a , and Claim-II follows. ◁

Then, fix an arbitrary $(u_i, v_j) \in \mathcal{E}_m$, we claim that the type-1 paths P_1 induce $a, b \in \Sigma$ such that $\pi_{u_i v_j}(a) = b$. Let p be the type-1 path that goes through v_j . For the path p to enter v_j , it must go through a one-cost arc (v_j^b, v_j) for some $b \in \Sigma$, and thus $b \in \sigma(v_j)$. Then, there are two ways to enter v_j^b :

1. from $w_{i'j}^{a'b}$ for some $(u_{i'}, v_j) \notin \mathcal{E}_m$ and $a' \in \pi_{u_{i'} v_j}^{-1}(b)$;
2. from w_{ij}^{ab} for $(u_i, v_j) \in \mathcal{E}_m$ and some $a \in \pi_{u_i v_j}^{-1}(b)$.

The first way is infeasible because $(u_{i'}^{a'}, w_{i'j}^{a'b}) \notin P_1$ due to Claim-I. Hence, the only way is the second and p must be exactly $r \rightarrow u_i^a \rightarrow w_{ij}^{ab} \rightarrow v_j^b \rightarrow v_j \rightarrow t_m$. Putting together Claim-II and the edge-disjointness of P_1 and P_{2-4} , there are $\deg_G(u_i) + 1$ arcs (r, u_i^a) in total in P . Thus the one-cost arc (r, u_i^a) must be included in $P \subseteq H$ because there are totally $\deg_G(u_i) + 1$ arcs (r, u_i^a) in G and thus $a \in \sigma(u_i)$. Therefore, we conclude that the arbitrarily fixed edge (u_i, v_j) is covered by $\pi_{u_i v_j}(a) = b$, where $a \in \sigma(u_i)$ and $b \in \sigma(v_j)$.

Hardness Gap

The one-to-one correspondence between solutions to the two problems is established by collecting the proofs for completeness and soundness. Furthermore, the reduction can be done in polynomial time in the size of the label cover instance and it guarantees that $|T| = \Delta$. Plugging in Corollary 8, the following inapproximability result for the k -DST problem is obtained.

► **Theorem 1.** For $k > |T|$, unless $\text{NP} = \text{ZPP}$, it is hard to approximate the k -DST problem to within a factor of $\Omega(|T|/\log |T|)$.

5 Inapproximability in Terms of the Connectivity Requirement

This section presents a hardness reduction, which is tailored for the approximation hardness in terms of the connectivity requirement k . Our reduction again takes a label cover instance $(\mathcal{G}, \Sigma, \pi)$ as an input and produces a k -DST instance $(k, G = (V, E), r, T)$. As we wish to obtain an inapproximability in terms of k , their main focus is in controlling the size of k .

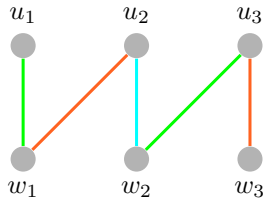
Base Construction

Our reduction starts from a basic building block.

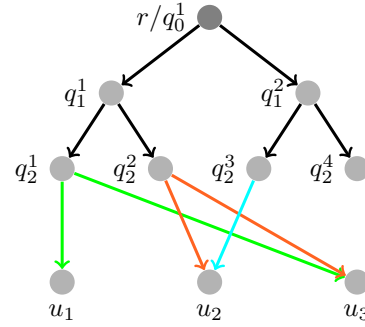
1. Let G be an empty graph. For each vertex $u_i \in \mathcal{U}$, create in G a counterpart of it (also named u_i) and a set of vertices $A_i = \{u_i^a : a \in \Sigma\}$; connect u_i to each $u_i^a \in A_i$ by an arc (u_i, u_i^a) of cost one.
2. For each vertex $v_j \in \mathcal{V}$, create in G a counterpart of it (also named v_j) and a set of vertices $B_j = \{v_j^b : b \in \Sigma\}$; connect each $v_j^b \in B_j$ to v_j by an arc (v_j^b, v_j) of cost one.
3. For each edge $(u_i, v_j) \in \mathcal{E}$, add to G a terminal t_{ij} and connect v_j to t_{ij} by a zero-cost arc (v_j, t_{ij}) . For $a, b \in \Sigma$, connect $u_i^a \in A_i$ to $v_j^b \in B_j$ by a zero-cost arc if $\pi_{u_i v_j}(a) = b$.

Gadget of d -ary Arborescence

All arcs created hereafter have zero cost. By Lemma 9, $\mathcal{E}$ is partitioned into $\delta \leq 2\Delta^2$ induced matchings $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_\delta$. Let $d \geq 2$ be an integral parameter to be determined later. We add to G a complete d -ary out-rooted tree (arborescence) Q of height $h = \lceil \log_d \delta \rceil$ with a rooted vertex r ; we set r as the root of the k -DST instance G . Choose an arbitrary order for the vertices in each layer of Q and use q_i^j to denote the j -th vertex of at the i -th layer. Note that q_0^1 is the root r . We then join each leaf q_h^j ($1 \leq j \leq \delta$) of Q to the counterpart in G of each vertex $u_i \in V(\mathcal{E}_j) \cap \mathcal{U}$. See Figure 8 and Figure 9 for an illustration. Intuitively, if we add a padding path that passes q_i^j , all illegal paths that go through the subtree rooted at q_i^j will get killed.



■ **Figure 8** A bipartite graph $\mathcal{G}$ with three induced matchings: $\mathcal{E}_1 = \{(u_1, w_1), (u_3, w_2)\}$, $\mathcal{E}_2 = \{(u_2, w_1), (u_3, w_3)\}$, $\mathcal{E}_3 = \{(u_2, w_2)\}$.



■ **Figure 9** The gadget when $d = 2$ with induced matchings $\mathcal{E}_1, \mathcal{E}_2$ and $\mathcal{E}_3$.

Final Construction

In the final construction, we add some *padding arcs* to the based graph. These arcs form *padding paths*, which then kill all the *illegal paths* through the help of the d -ary arborescence.

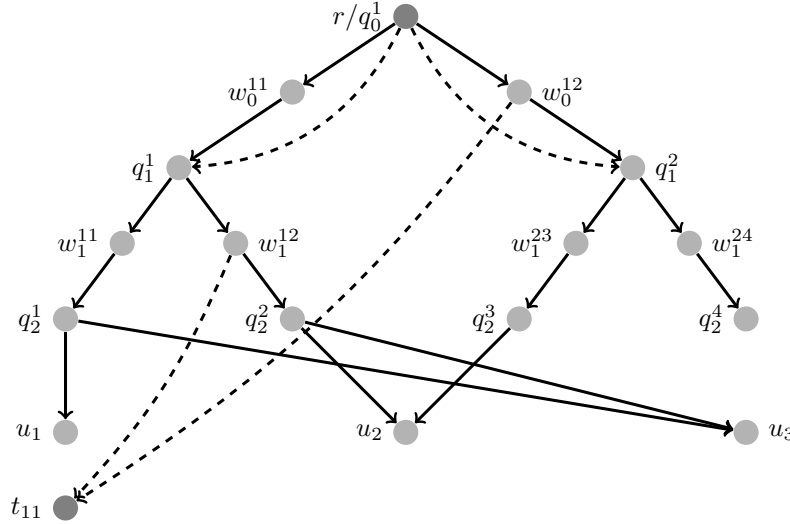
1. For each arc $(q_i^j, q_{i+1}^{j'})$ of Q , we replace it by two arcs $(q_i^j, w_i^{jj'})$ and $(w_i^{jj'}, q_{i+1}^{j'})$.

We still use Q to denote the original arborescence for notational convenience.

2. For each $1 \leq i \leq h-1$ and j , we add $d-1$ copies of an arc (r, q_i^j) to G .
3. (Refer to Figure 10) For each terminal t_{ij} , suppose the edge (u_i, v_j) is in the group $\mathcal{E}_m$ (corresponding to q_h^m). There is a unique path in the d -ary arborescence from r to q_h^m : $(r = q_0^{j_0=1}) \rightarrow q_1^{j_1} \rightarrow q_2^{j_2} \rightarrow \dots \rightarrow q_h^{j_h=m}$. At each level $1 \leq \ell \leq h$, there are $d-1$ siblings of $q_\ell^{j_\ell}$, and for each sibling $q_\ell^{j \neq j_\ell}$, we add a zero-cost arc from $w_{\ell-1}^{j_{\ell-1}j}$ to the terminal t_{ij} . After the construction, the in-degree of each terminal t_{ij} is $h(d-1)+1$ and we set k as it.

Why Does it Work under Grouping by Induced Matchings?

For any edge $(u_i, v_j) \in \mathcal{E}_m$, now we can promise that there must be a path from r to t_{ij} going through the leaf q_h^m and the vertex $u_i \in V(G)$. Let us examine if illegal paths are bypassing either q_h^m or u_i . The padding arcs can kill illegal paths from r to t_{ij} that depart from the unique path $r \rightarrow q_1^{j_1} \rightarrow q_2^{j_2} \rightarrow \dots \rightarrow q_h^{j_h=m} \rightarrow u_i$ at some vertex $q_\ell^{j_\ell}$ for $\ell < h$. It remains to check if there is an illegal path of the form $r \rightarrow q_1^{j_1} \rightarrow q_2^{j_2} \rightarrow \dots \rightarrow q_h^{j_h=m} \rightarrow u_{i'} \rightarrow \dots \rightarrow v_j \rightarrow t_{ij}$ for some $u_{i'} \neq u_i$. The existence of such $u_{i'}$ implies that $(u_{i'}, v_j) \in \mathcal{E}_m$, which contradicts that $\mathcal{E}_m$ is an induced matching.



■ **Figure 10** Padding arcs for the terminal t_{11} are dashed. Note that the vertices u_i have outgoing arcs, but not drawn here.

Completeness

Given a feasible multilabeling σ of the instance $(\mathcal{G}, \Sigma, \pi)$, we show that there is a corresponding k -connected subgraph $H = (V, F)$ of $G = (V, E)$ such that $c(\sigma) = c(H)$. The set F consists of three types of arcs: 1) all zero-cost arcs in G ; 2) the one-cost arcs (u_i, u_i^a) for each $u_i \in \mathcal{U}$ and $a \in \sigma(u_i)$; 3) the one-cost arcs (v_j^b, v_j) for each $v_j \in \mathcal{V}$ and $b \in \sigma(v_j)$. Clearly, $c(H) = c(\sigma)$ and the definition of F induces an injective mapping.

Now we prove that for any terminal $t_{ij} \in T$ there are k edge-disjoint paths, denoted by P , in H from r to t_{ij} . Let $\mathcal{E}_m$ be the induced matching that contains t_{ij} . There is a unique path $p = (r = q_0^{j_0=1}) \rightarrow q_1^{j_1} \rightarrow \dots \rightarrow q_h^{j_h=m}$ in the arborescence Q from r to q_h^m . Since $\text{indeg}_H(t_{ij}) = k$, each arc entering t_{ij} must be contained in P . We consider two cases:

1. (v_j, t_{ij}) : The feasibility of σ induces $a \in \sigma(u_i)$ and $b \in \sigma(v_j)$ such that $\pi_{u_i v_j}(a) = b$, so that $(u_i, u_i^a), (u_i^a, v_j^b), (v_j^b, v_j) \in F$. Guided by the path p , we add to P the path $r \rightarrow w_0^{j_0 j_1} \rightarrow q_1^{j_1} \rightarrow w_1^{j_1 j_2} \rightarrow \dots \rightarrow w_{h-1}^{j_{h-1} j_h = m} \rightarrow q_h^{j_h = m} \rightarrow u_i \rightarrow u_i^a \rightarrow v_j^b \rightarrow v_j \rightarrow t_{ij}$.
2. $(w_\ell^{j_\ell j'}, t_{ij})$: Here $j' \neq j_{\ell+1}$. We add to P the path $r \rightarrow q_\ell^{j_\ell} \rightarrow w_\ell^{j_\ell j'} \rightarrow t_{ij}$.

For $0 \leq \ell < h$, the type-1 path goes through $q_\ell^{j_\ell} \rightarrow w_\ell^{j_\ell j_{\ell+1}} \rightarrow q_{\ell+1}^{j_{\ell+1}}$, while each type-2 path goes through $r \rightarrow q_\ell^{j_\ell} \rightarrow w_\ell^{j_\ell j'} \rightarrow t_{ij}$ for some j' . Since $j' \neq j_{\ell+1}$, the type-1 path and the type-2 paths do not share any common arc. For the type-2 paths themselves, for each $0 \leq \ell < h$ they consume in total $d-1$ duplicates of the arc $(r, q_\ell^{j_\ell})$, which is prepared well in the second step of the final construction of G .

Soundness

Given a k -connected subgraph $H = (V, F)$ (that contains all zero-cost arcs) of the k -DST instance (k, G, r, T) , we show that there is a corresponding feasible multilabeling σ of the label cover instance $(\mathcal{G}, \Sigma, \pi)$ such that $c(\sigma) = c(H)$. We define σ as follows: set $\sigma(u_i)$ as $\{a \in \Sigma : (u_i, u_i^a) \in F\}$ for $u_i \in \mathcal{U}$ and set $\sigma(v_j)$ as $\{b \in \Sigma : (v_j^b, v_j) \in F\}$ for $v_j \in \mathcal{V}$. Clearly, $c(\sigma) = c(H)$ and the definition of σ induces an injective mapping. Then we prove that σ covers all the edges in $\mathcal{E}$.

Consider an edge $(u_i, v_j) \in \mathcal{E}_m$ and its corresponding terminal t_{ij} . Let $(r = q_0^{j_0=1}) \rightarrow q_1^{j_1} \rightarrow \dots \rightarrow q_h^{j_h=m}$ be the unique path in the arborescence Q from r to q_h^m . Let P be any set of k edge-disjoint paths from r to t_{ij} in H . The fact that $\text{indeg}_G(t_{ij}) = k = h(d-1) + 1$ forces P to contain all arcs entering t_{ij} . These arcs are of two types:

1. One arc (v_j, t_{ij}) ;
2. $h(d-1)$ arcs of $(w_\ell^{j_\ell j'}, t_{ij})$ for $0 \leq \ell < h$ and $j' \neq j_{\ell+1}$.

Let P_1 be the only path in P that uses the type-1 arc, and let P_2 be the union of paths in P that use type-2 arcs. By backtracking the paths in P_2 from t_{ij} for two steps, it holds, for $0 \leq \ell < h$ and $j' \neq j_{\ell+1}$, that $(q_\ell^{j_\ell}, w_\ell^{j_\ell j'}) \in P_2$. Thus, the path P_1 has to be $q_0^{j_0=1} \rightarrow w_0^{j_0 j_1} \rightarrow q_1^{j_1} \rightarrow \dots \rightarrow q_h^m \rightarrow \dots \rightarrow v_j \rightarrow t_{ij}$. Let us backtrack P_1 from v_j . The previous vertex must be v_j^b for some $b \in \Sigma$, and u_i^a for some $a \in \Sigma$, and $u_{i'}$, and then q_h^m . If $i' \neq i$, then by the construction of G , we know that $u_{i'} \in V(\mathcal{E}_m)$. However, it also holds that $v_j \in V(\mathcal{E}_m)$ because $(u_i, v_j) \in \mathcal{E}_m$. Thus, the induced subgraph on $V(\mathcal{E}_m)$ contains both (u_i, v_j) and $(u_{i'}, v_j)$, contradicting the matching property of $\mathcal{E}_m$. Therefore, $i' = i$, implying that $a \in \sigma(u_i)$, $b \in \sigma(v_j)$ and $\pi_{u_i v_j}(a) = b$ because $(u_i, u_i^a), (u_i^a, v_j^b), (v_j^b, v_j) \in P_1 \subseteq P \subseteq H$.

Hardness Gap

In this subsection, we deduce approximation hardness for k -DST. Clearly, the reduction can be made in polynomial time in the size of the input label cover instance. By setting different values of the parameter d , we have the following two propositions.

► **Theorem 3.** *It is hard to approximate the k -DST problem on L -layered graphs $G = (V, E)$ for $\Omega(1) \leq L \leq O(\log |V|)$ to within a factor of $\Omega\left((k/L)^{(1-\epsilon)L/4-2}\right)$ for any constant $\epsilon > 0$, unless $\text{NP} = \text{ZPP}$.*

Proof. Let L be the height (the maximum length of paths) of the underlying graph in the k -DST instance. Recall that in the construction we have a modified d -ary arborescence and 5 base levels with vertices u_i, u_i^a, v_j^b, v_j and t_{ij} . So, $L = 2\lceil \log_d \delta \rceil + 5$. By Lemma 9, δ is at most $2\Delta^2$. Thus, $L \leq 2\lceil \log_d(2\Delta^2) \rceil + 5$. To complete the proof, we add some dummy vertices to the modified d -ary arborescence so that the height becomes exactly $2\lceil \log_d(2\Delta^2) \rceil + 5$.

We fix $d = \Delta^x \geq 2$ for some $0 < x \leq 1$. Then $k = \lceil \log_d \delta \rceil (d - 1) + 1 \leq (d - 1)(2 \log \Delta / \log d + 1 / \log d + 1) + 1 \leq (2/x + 2) \Delta^x$. We also have that $4/x \leq L = 2 \lceil \log_d (2\Delta^2) \rceil + 5 \leq 4/x + 8$. Therefore, $(k/L)^{\frac{(1-\epsilon)L}{4}-2} \leq \Delta^{1-\epsilon}$, and we have the claimed result on layered graphs by plugging in Corollary 8. The parameter x can be used to obtain specific values of L that we want. ◀

► **Theorem 4.** *For $k < |T|$, it is hard to approximate the k -DST problem to within a factor of $\Omega(2^{k/2}/k)$, unless $\text{NP} = \text{ZPP}$.*

Proof. Recall that $k = h(d - 1) + 1 = \lceil \log_d \delta \rceil (d - 1) + 1$ where $\delta \leq 2\Delta^2$. If we set $d = 2$, then $k = \lceil \log \delta \rceil + 1 \leq 2 \log \Delta + 3$. If k is too small, then we can add dummy arcs from r to terminals to make $k \geq \log \Delta$. Plugging in Corollary 8, the claimed hardness gap follows. ◀

6 Discussion

In this paper we obtain improved inapproximability results for the k -DST problem in terms of parameters k and $|T|$. All of the results are derived from the same hardness source, the (minimum) label cover problem, which admits no $\Delta / \log \Delta$ -approximation algorithm under standard complexity assumptions, where Δ is the maximum degree of the underlying bipartite graph in the label cover problem. The label cover problem also admits no $2^{\log^{1-\epsilon} N}$ -approximation algorithm for any constant $\epsilon > 0$, where N is the underlying graph size. Note that $2^{\log^{1-\epsilon} N} = N^{o(1)}$. If we apply the reduction in the $\Omega(|T| / \log |T|)$ -hardness, then it transfers to $k^{o(1)}$ which is even worse than the previous $\Omega(k / \log k)$ -hardness, not to say the factor of $\Omega(2^{k/2}/k)$ in this paper. If we apply the reduction in the $\Omega(2^{k/2}/k)$ -hardness, then it transfers to $|T|^{o(1)}$ which is still much worse than $|T|^{1/4-\epsilon}$ and $\Omega(|T| / \log |T|)$.

References

- 1 Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193. IEEE Computer Society, 1996. doi:10.1109/SFCS.1996.548477.
- 2 Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoss, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1), February 2013. doi:10.1145/2432622.2432628.
- 3 Parinya Chalermsook, Syamantak Das, Guy Even, Bundit Laekhanukit, and Daniel Vaz. Survivable network design for group connectivity in low-treewidth graphs. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 8:1–8:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.8.
- 4 Parinya Chalermsook, Syamantak Das, Bundit Laekhanukit, and Daniel Vaz. Beyond metric embedding: Approximating group steiner trees on bounded treewidth graphs. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 737–751. SIAM, 2017. doi:10.1137/1.9781611974782.47.
- 5 Parinya Chalermsook, Fabrizio Grandoni, and Bundit Laekhanukit. On survivable set connectivity. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 25–36. SIAM, 2014.
- 6 Parinya Chalermsook, Fabrizio Grandoni, and Bundit Laekhanukit. On survivable set connectivity. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 25–36. SIAM, 2015. doi:10.1137/1.9781611973730.3.

- 7 Chun-Hsiang Chan, Bundit Laekhanukit, Hao-Ting Wei, and Yuhao Zhang. Polylogarithmic approximation algorithm for k-connected directed steiner tree on quasi-bipartite graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176, pages 63:1–63:20. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.
- 8 Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. *Journal of Algorithms*, 33(1):73–91, 1999.
- 9 Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. Rounding via trees: Deterministic approximation algorithms for group steiner trees and k -median. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 114–123. ACM, 1998. doi:10.1145/276698.276719.
- 10 Chandra Chekuri, Guy Even, and Guy Kortsarz. A greedy approximation algorithm for the group steiner problem. *Discret. Appl. Math.*, 154(1):15–34, 2006. doi:10.1016/j.dam.2005.07.010.
- 11 Joseph Cheriyan, Bundit Laekhanukit, Guylain Naves, and Adrian Vetta. Approximating rooted steiner networks. *ACM Transactions on Algorithms (TALG)*, 11(2):1–22, 2014.
- 12 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633. ACM, 2014. doi:10.1145/2591796.2591884.
- 13 Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004. doi:10.1016/j.jcss.2004.04.011.
- 14 Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.
- 15 Lisa Fleischer, Kamal Jain, and David P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences*, 72(5):838–867, 2006. Special Issue on FOCS 2001.
- 16 András Frank. Rooted k -connections in digraphs. *Discret. Appl. Math.*, 157(6):1242–1254, 2009. doi:10.1016/j.dam.2008.03.040.
- 17 András Frank and Éva Tardos. Generalized polymatroids and submodular flows. *Math. Program.*, 42(1-3):489–563, 1988. doi:10.1007/BF01589418.
- 18 Zachary Friggstad, Jochen Könnemann, and Shadravan Mohammad. A Logarithmic Integrality Gap Bound for Directed Steiner Tree in Quasi-bipartite Graphs. In Rasmus Pagh, editor, *15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016)*, volume 53 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:11, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- 19 Naveen Garg, Goran Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000. doi:10.1006/jagm.2000.1096.
- 20 Rohan Ghuge and Viswanath Nagarajan. Quasi-polynomial algorithms for submodular tree orienteering and other directed network design problems. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1039–1048. SIAM, 2020. doi:10.1137/1.9781611975994.63.
- 21 Michel X Goemans, Neil Olver, Thomas Rothvoß, and Rico Zenklusen. Matroids and integrality gaps for hypergraphic steiner tree relaxations. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1161–1176. SIAM, 2012.
- 22 Fabrizio Grandoni and Bundit Laekhanukit. Surviving in directed graphs: a quasi-polynomial-time polylogarithmic approximation for two-connected directed steiner tree. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 420–428, 2017.

- 23 Fabrizio Grandoni, Bundit Laekhanukit, and Shi Li. $O(\log^2 k / \log \log k)$ -approximation algorithm for directed Steiner tree: a tight quasi-polynomial-time algorithm. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 253–264, 2019.
- 24 Anupam Gupta, Ravishankar Krishnaswamy, and R. Ravi. Tree embeddings for two-edge-connected network design. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1521–1538. SIAM, 2010. doi:10.1137/1.9781611973075.124.
- 25 Eran Halperin, Guy Kortsarz, Robert Krauthgamer, Aravind Srinivasan, and Nan Wang. Integrality ratio for group steiner trees and directed steiner trees. *SIAM J. Comput.*, 36(5):1494–1511, 2007. doi:10.1137/S0097539704445718.
- 26 Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. In Lawrence L. Larmore and Michel X. Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 585–594. ACM, 2003. doi:10.1145/780542.780628.
- 27 Tomoya Hibi and Toshihiro Fujito. Multi-rooted greedy approximation of directed steiner trees with applications. *Algorithmica*, 74(2):778–786, 2016.
- 28 Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.
- 29 Rohit Khandekar, Guy Kortsarz, and Zeev Nutov. Approximating fault-tolerant group-steiner problems. *Theor. Comput. Sci.*, 416:55–64, 2012. doi:10.1016/j.tcs.2011.08.021.
- 30 Samir Khuller and Uzi Vishkin. Biconnectivity approximations and graph carvings. *J. ACM*, 41(2):214–235, 1994. doi:10.1145/174652.174654.
- 31 Guy Kortsarz, Robert Krauthgamer, and James R. Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM J. Comput.*, 33(3):704–720, 2004. doi:10.1137/S0097539702416736.
- 32 Bundit Laekhanukit. Parameters of two-prover-one-round game and the hardness of connectivity problems. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1626–1643. SIAM, 2014.
- 33 Bundit Laekhanukit. Approximating directed steiner problems via tree embedding. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.
- 34 Pasin Manurangsi. A note on degree vs gap of min-rep label cover and improved inapproximability for connectivity problems. *Information Processing Letters*, 145:24–29, 2019.
- 35 Pasin Manurangsi, Aviad Rubinfeld, and Tselil Schramm. The strongish planted clique hypothesis and its consequences. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 10:1–10:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.10.
- 36 Zeev Nutov. Approximating minimum-cost connectivity problems via uncrossable bifamilies. *ACM Transactions on Algorithms (TALG)*, 9(1):1–16, 2012.
- 37 Zeev Nutov. On rooted k -connectivity problems in quasi-bipartite digraphs. In *International Computer Science Symposium in Russia*, pages 339–348. Springer, 2021.
- 38 Alexander Zelikovsky. A series of approximation algorithms for the acyclic directed steiner tree problem. *Algorithmica*, 18(1):99–110, 1997.