Length-Constrained Network Design in Planar Digraphs

Chandra Chekuri
Siebel School of Computing and Data Science, University of Illinois, Urbana-Champaign, IL, USA

Rhea Jain
Siebel School of Computing and Data Science, University of Illinois, Urbana-Champaign, IL, USA

Abstract

We study *length-constrained* generalizations of Directed Steiner Tree (DST) and Directed Steiner Forest (DSF) in planar digraphs. In both problems, the input is a directed graph with edge costs. DST asks for a min-cost subgraph connecting a root to a given set of terminals, and DSF asks for a min-cost subgraph connecting each of a given set of source-sink terminal pairs. In the length-constrained setting, each edge has both a cost and a length, and the input includes a length bound h; the goal is to find a min-cost subgraph connecting each terminal pair via a path of length at most h. Our work is motivated by a recent line of results showing that several network design problems that are traditionally hard in directed graphs admit polylogarithmic approximation ratios in planar digraphs. We give polylogarithmic bicriteria approximation algorithms for length-constrained analogues of DST and DSF in planar digraphs. Our approximation ratios match the best known for DST and DSF in planar digraphs, with an $O(\log k)$ violation of the length constraint, where k denotes the number of terminals (or terminal pairs). As corollaries, we obtain polylogarithmic approximations for *buy-at-bulk* DST and DSF in planar digraphs.

2012 ACM Subject Classification Theory of computation → Routing and network design problems; Theory of computation → Discrete optimization

Keywords and phrases Approximation algorithms, network design, directed graphs, length constraints

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.15

Category APPROX

Related Version *Full Version*: https://arxiv.org/abs/2607.25811

Funding Chandra Chekuri and Rhea Jain are partially supported by NSF grant CCF-2402667.

Acknowledgements We thank David Zheng for helpful discussion on length-constrained Directed Steiner Forest.

1 Introduction

Directed Steiner Tree (DST) and *Directed Steiner Forest* (DSF) are fundamental problems in network design with a wide array of applications. For both problems, the input consists of a directed graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$. In Directed Steiner Tree, given a root $s \in V$ and terminals $S \subseteq V \setminus \{s\}$, the goal is to find a min-cost subgraph containing an s-t path for each $t \in S$. Directed Steiner Forest generalizes DST to the *multicommodity* setting: given a set of demand pairs $D = \{(s_i, t_i)\}_{i \in [k]}$, the goal is to find a min-cost subgraph containing an s_i-t_i path for each $i \in [k]$.

DST and DSF are NP-hard, and in fact have strong lower bounds on their approximability. DST generalizes several hard problems in combinatorial optimization, such as Set Cover and Group Steiner Tree. It is hard to approximate to an $\Omega(\log^2 k/\log\log k)$-factor under plausible complexity assumptions [33], and to a slightly weaker $\Omega(\log^{2-\epsilon} k)$-factor unless

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2026).
Editors: Mohit Singh and Tom Gur; Article No. 15; pp. 15:1–15:24
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

NP is contained in randomized quasi-poly time [37]; here, k denotes $|S|$. The best known approximation ratios are $O(\log^2 k/\log\log k)$ in *quasi-polynomial* time [33, 31, 10] and $O(k^\epsilon)$ for any fixed $\epsilon > 0$ in polynomial time [60]. Whether DST admits a polylogarithmic approximation in polynomial time is a major open question. We remark that the natural LP relaxation for DST has polynomial-factor integrality gap [61, 51]; this is one reason that developing improved approximations for DST has proven challenging. DSF is hard to approximate to a factor $\Omega(2^{\log^{1-\epsilon}(n)})$ for any $\epsilon > 0$ and thus does not admit a polylogarithmic approximation [24]. The best known approximation ratios are $O(k^{\frac{1}{2}+\epsilon})$ [12] and $O(n^{2/3+\epsilon})$ [7]. An exciting recent line of work has shown that many problems that are hard to approximate in general directed graphs are tractable when the input graph is *planar* [45, 29, 17, 14]. In planar digraphs, DST admits an $O(\log k)$-approximation [29], a natural LP relaxation has integrality gap $O(\log^2 k)$ [17], and DSF admits an $O(\log^6 k)$-approximation [14].

In this paper, we consider a generalization of DST and DSF to the *length-constrained* setting. As the name suggests, these impose an additional constraint on the length of paths used to connect terminal pairs. Formally, in *Length-Constrained Directed Steiner Tree* (LC-DST) and *Length-Constrained Directed Steiner Forest* (LC-DSF), we are given an additional *length* function on the edges $\ell : E \to \mathbb{R}_+$ and a length constraint h; the goal is to find a min-cost subgraph in which terminal pairs are connected via a path of length at most h. There are several practical reasons why connectivity alone may not suffice: short paths are desirable for faster transportation and communication, improved tolerance to failure, and more. Furthermore, length-constrained network design is closely related to several other important problems in combinatorial optimization. For instance, *buy-at-bulk* network design (closely related to cost-distance and other problems) is a well-studied and practical model which places a penalty on the path lengths in the objective function instead of imposing strict bounds on path lengths. Progress in length-constrained network design has led to the resolution of several open problems in buy-at-bulk network design [13, 16]. As a result of its practical and theoretical importance, length-constrained network design has been very well studied; see Section 1.3 for details.

In undirected graphs, adding length constraints to problems adds significant complexity to a problem. As an example, consider the minimum spanning tree problem; this is easily solvable in polynomial time. However, the length-constrained minimum spanning tree problem is essentially equivalent in approximability to DST (see [39] for a formal reduction). A common way to handle this difficulty is by relaxing the length constraint, allowing for some slack in both the length and the cost (see Definition 1). In directed graphs, on the other hand, length constraints do not add inherent complexity. A simple reduction shows that LC-DST is equivalent to DST, as shown in Figure 1,[1] and one can define a similar reduction from LC-DSF to DSF. However, these reductions do not preserve planarity. Given the fruitful progress on network design problems in planar digraphs, it is natural to ask if LC-DST and LC-DSF admit polylogarithmic approximations in planar digraphs. We answer these questions affirmatively.

1.1 Our Results

We obtain polylogarithmic *bicriteria* approximation algorithms for LC-DST and LC-DSF in planar digraphs.

[1] For polynomial edge lengths one can replace each edge e with a path of length $\ell(e)$. For larger edge lengths, one can obtain a similar reduction losing a $(1 + \epsilon)$ factor in length slack.

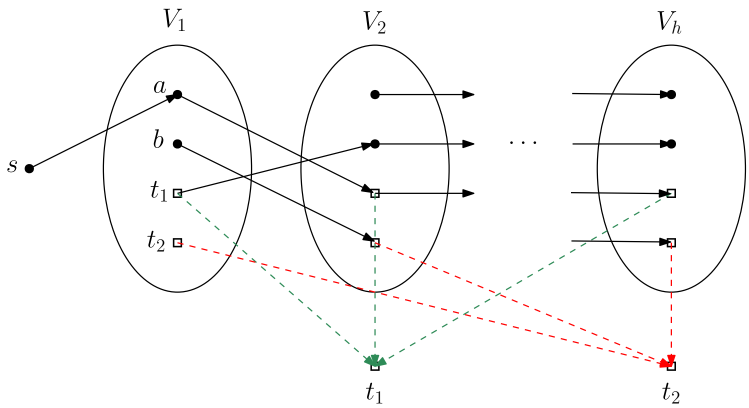


Figure 1 This example assumes all edge lengths are 1. Suppose the input graph is a path s, a, t_1, b, t_2 with terminals t_1, t_2. We create h copies of the vertex set $V_1, \ldots, V_h$, and for every $(u, v) \in E$ and all $i < h$, we add an edge from the copy of u in V_i to that of v in V_{i+1}. The terminals in the resulting DST instance are given at the bottom of the figure.

▶ **Definition 1.** *An (α, β)-bicriteria approximation for a length constrained network design problem with length constraint h is a solution $F \subseteq E$ in which all demand pairs are connected via a path of length at most $\beta \cdot h$, and $c(F) \leq \alpha \cdot \mathrm{OPT}$, where* OPT *is the cost of an optimal solution. We refer to α as the* approximation ratio *and β as the* length slack*.*

The approximation factors for all of our results match the best known ratios for DST and DSF in planar digraphs without length constraints; all algorithms incur an $O(\log k)$ factor in the length slack. We remark that this $O(\log k)$ length slack factor is essentially the best known for many problems in length-constrained network design when one seeks a polylogarithmic approximation ratio, including length-constrained MST in undirected graphs. There are known algorithms for length-constrained MST with polynomial approximation ratio that do not lose any length slack. Reducing the length slack to $O(1)$ while ensuring polylogarithmic approximation ratio is an enticing research direction with some partial progress discussed further in Section 1.3. We highlight one such recent work: in undirected planar graphs, [39] obtains an $O(\log^{1+\epsilon}(n))$-approximation with $(1+\epsilon)$ length slack. However, it is unknown if this approach would extend to the directed setting, and there appear to be several challenges.

We first consider length-constrained Directed Steiner Tree. An approximation algorithm is *LP-competitive* if the cost of the given solution is at most α times the value of an optimum *fractional* solution to a natural LP relaxation. We describe an LP relaxation to LC-DST in Section 3. We obtain two algorithms: the first (Theorem 2) is with respect to the optimum integral solution, and the second (Theorem 3) is LP-competitive.

▶ **Theorem 2.** *There is a bicriteria $(O(\log k), O(\log k))$-approximation for LC-DST in planar digraphs.*

▶ **Theorem 3.** *There is a bicriteria $(O(\log^2 k), O(\log k))$ LP-competitive approximation for LC-DST in planar digraphs.*

Next, we consider length-constrained Directed Steiner Forest. Note that Theorem 4 provides an approximation algorithm with respect to the optimum integral solution; obtaining an LP-competitive approximation remains open. We note that an LP-competitive approximation for planar DSF is not known even without length constraints.

▶ **Theorem 4.** *There is a bicriteria $(O(\log^6 k), O(\log k))$-approximation for Length-Constrained Directed Steiner Forest in planar digraphs.*

1.1.1 Extensions

We mention two further extensions of our work. Details and formal theorem statements are omitted as they are technically involved and require new machinery, see [41] for details.

Buy-at-bulk network design. As mentioned earlier, there are known connections between buy-at-bulk and length-constrained network design problems. Prior work in *undirected* graphs [13, 16] demonstrated an LP-based reduction from a buy-at-bulk problem to its corresponding length-constrained problem. The resulting approximation ratio for the buy-at-bulk problem depends on both the approximation and length slack of the algorithm for the length-constrained problem. These arguments extend to directed graphs; thus, one can obtain a polylogarithmic approximation for buy-at-bulk DST in planar digraphs as a corollary of Theorem 3. In the multicommodity setting, more effort is needed, since Theorem 4 does not yield an LP-competitive solution. Nevertheless, one can obtain a polylogarithmic approximation for buy-at-bulk DSF in planar digraphs by arguing that there exists a low-density *junction tree*. Similar arguments have been used in [11, 4, 13].

Group, Covering, and Polymatroid Steiner Tree. Chekuri et al. [17] developed polylogarithmic approximation algorithms for several rooted generalizations of Directed Steiner Tree in planar digraphs. This includes the *Directed Polymatroid Steiner Tree* problem, where a polymatroid is defined on the vertex set, and a feasible tree is required to span a base of the polymatroid. A polymatroid is an integer-valued, normalized, monotone, submodular function. This generalizes several important network design problems, including Group and Covering Steiner Tree. At a high level, the algorithm of [17] explicitly constructs the recursion tree of the divide-and-conquer algorithm of [29]. One can view this recursion tree as a "tree embedding" suitable for solving rooted problems, allowing one to reduce to solving the problem on a tree. One can similarly construct a "tree embedding" that respects length constraints, based on the divide-and-conquer algorithm of Theorem 2. Thus, one can obtain polylogarithmic approximations for length-constrained rooted variants of directed polymatroid, covering, and group problems.

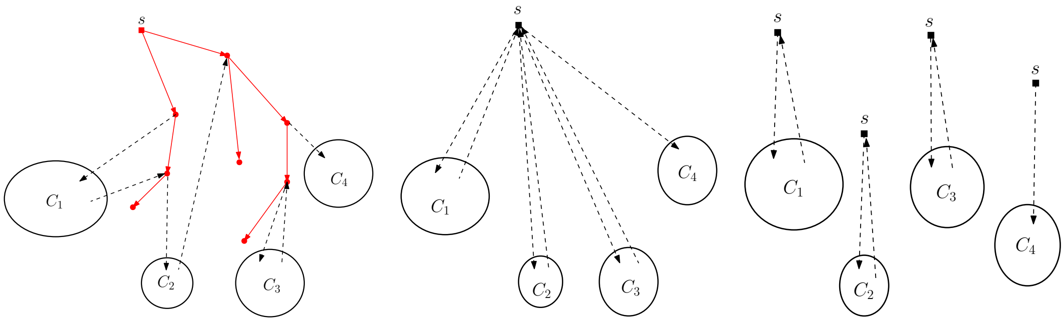
1.2 Technical Overview

The proofs of Theorem 2 and Theorem 3 on LC-DST adapt established ideas to the length-constrained setting. The main novel contribution of this work is Theorem 4 on LC-DSF, as the previous approach for DSF in planar digraphs fails to extend. We outline the ideas for all three algorithms here.

Approximation for DST in planar digraphs. We start by giving an overview of the divide-and-conquer $O(\log k)$-approximation given by Friggstad and Mousavi [29]. The algorithm uses Thorup's shortest path separator theorem applied to directed graphs:

▶ **Lemma 5** ([29, 58]). *Let G be a planar directed graph with non-negative edge costs $c(e)$, non-negative vertex weights $w(v)$, and a root $s \in V$ such that every vertex in V is reachable from s. There exists a polynomial time algorithm to find three shortest dipaths P_1, P_2, P_3 starting at s such that every weakly connected component of $G \setminus (P_1 \cup P_2 \cup P_3)$ has at most half the vertex weight of G.*

The high-level idea in [29] is simple; see Figure 2. Suppose we can guess the optimum solution value OPT for a given DST instance. Then, we can remove all vertices farther than OPT from s and use Lemma 5 to find three paths, each of cost at most OPT, whose



(a) Step 1. **(b)** Step 2. **(c)** Step 3.

Figure 2 High-level overview of the divide-and-conquer algorithm. In Step 1, we compute the planar separator (shown in red). Step 2 demonstrates contracting the separator into the root, and Step 3 demonstrates the resulting independent sub-instances.

removal results in connected components each containing at most half the terminals. We shrink the paths into s and recurse on the resulting sub-instances. The recursion depth is $O(\log k)$, which bounds the total cost to $O(\log k) \cdot \mathrm{OPT}$. Implementing the guess of OPT in each recursive call is challenging; naively guessing requires quasipolynomial time. [29] obtain a polynomial time algorithm by a refined argument where the guess for OPT is folded into the recursion itself.

Handling Length-Constraints and Bounding Integrality Gap. The main challenge in using the approach of [29] in *length-constrained* Directed Steiner Tree lies in defining the shortest path tree on which to apply the planar separator result. Ideally, one would like for the tree to capture length-constrained distances, where the h-length-constrained distance from u to v is the min-cost over all u-v paths of length at most h. Unfortunately, length-constrained distances do not form a metric, so one cannot use a standard shortest path tree. To circumvent this issue, we use the well-known "mixture-metric", which balances length constraints and costs. This suffices to obtain Theorem 2. We remark that recent work of [39] on length-constrained MST in *undirected planar graphs* also uses the mixture-metric to apply the separator lemma. To prove Theorem 3, we use an idea introduced in [17], which uses the LP optimum value as the estimate for OPT instead of guessing. The algorithm of [17] extends to the length-constrained setting with the mixture-metric in a similar fashion.

Directed Steiner Forest via Junction Trees. To tackle the multicommodity setting, we build on the *junction tree* based algorithm of [14] for planar DSF. At a high level, junction trees allow one to reduce a multicommodity problem to its single-source counterpart. This technique has been used in a variety of network design problems; it was first explicitly described in the context of non-uniform buy-at-bulk network design [36, 11], though it was implicitly used earlier for Directed Steiner Forest in [10]. The idea is to repeatedly find well-structured *low-density partial solutions*, where a partial solution is feasible for a subset of terminal pairs, and the density is the ratio of cost to the number of terminal pairs connected. Using a standard iterative approach for covering problems, one can reduce a multicommodity problem to its corresponding min-density partial solution problem, losing an $O(\log k)$ factor in the approximation ratio. We aim to find partial solutions that contain a "junction" vertex $v \in V$ through which many pairs connect. This allows us to view the problem as a single-source problem with source v. The approximation for DSF in planar digraphs of [14] consists of two steps: (1) proving the existence of a low-density junction tree, and (2) efficiently finding an approximate min-density junction tree. The second step uses a bucketing-and-scaling technique introduced by [11], which relies on the existence of an LP-competitive single-source algorithm. With Theorem 3, this extends with little difficulty to the length-constrained setting. Our main technical contribution is the first step: showing existence of a low-density junction tree, as length constraints pose several new challenges, and the proof of [14] does not cleanly extend. We defer further discussion to Section 4.

1.3 Related Work

Directed Steiner Tree. Zelikovsky [60] was the first to address the approximability of DST. He obtained an $O(k^\epsilon)$-approximation for any fixed $\epsilon > 0$ via two ideas. He developed a recursive greedy algorithm and analyzed its performance as a function of recursion depth. He then showed that one can reduce the problem on a general directed graph to a problem on a depth d DAG (via the transitive closure of the original graph) at the loss of an approximation factor that depends on d. Charikar et al. [10] refined the algorithm and analysis in [60] and combined it with the depth reduction. They showed that one can obtain an $O(d^2 k^{1/d} \log k)$ approximation in $O(n^{O(d)})$-time; this led to an $O(\log^3 k)$-approximation in quasi-polynomial time. Subsequently, Grandoni et al. [33] improved the approximation to $O(\log^2 k / \log\log k)$ in quasi-polynomial time via a more sophisticated LP-based approach. A different approach that also yields the same bound was given by Ghuge and Nagarajan [31], and this is based on a refinement of the recursive greedy algorithm for walks in graphs [18]. The approach of [31] extends to several related problems, including directed single-source buy-at-bulk.

Zosin and Khuller [61] showed that the natural cut-based LP relaxation for DST has an integrality gap of $\Omega(\sqrt{k})$. However, their example only showed a gap of $\Omega(\log n)$ as a function of the number of nodes n. There was some hope that the integrality gap is polylogarithmic in n; however, [51] recently showed that the gap is $\Omega(n^\delta)$ for some $\delta > 0$ by modifying the construction in [61]. Interestingly, these lower bound examples are DAGs with $O(1)$-layers, for which the recursive-greedy algorithm yields an $O(\log k)$-approximation in polynomial-time. Rothvoss [57] showed that $O(\ell)$-levels of the Lasserre SDP hierarchy when applied to the standard cut-based LP reduces the integrality gap to $O(\ell \log k)$ on DAGs with ℓ layers. This was later refined to show that $O(\ell)$-levels of the Sherali-Adams hierarchy suffices [27]. However, both these approaches also require quasi-polynomial time to obtain a polylogarithmic approximation. Recent work by Laekhanukit [50] shows a flow-based LP relaxation that has polylogarithmic integrality gap under some strong assumptions on the structure of an optimal fractional solution.

Directed Steiner Forest. The first nontrivial approximation for Directed Steiner Forest was an $\tilde{O}(k^{2/3})$-approximation given by Charikar et al. [10]. This follows a similar iterative density-based procedure as the *junction tree* approach; however, they restrict to trees of a much simpler structure. This approximation ratio was subsequently improved to $O(k^{\frac{1}{2}+\epsilon})$ by Chekuri et al. [12]; they showed that given an instance (G, D) of DSF, there exists a junction tree of density at most $O(k^{1/2})$ times the optimum. They then provided an algorithm to find a low-density junction tree via height reduction and Group Steiner Tree rounding. DSF has improved approximation ratios when k is large. Feldman, Kortsarz, and Nutov [25] obtained an $O(n^\epsilon \cdot \min(n^{4/5}, m^{2/3}))$-approximation using a junction-based approach. This analysis was refined by Berman et al. [7] using ideas developed for finding directed spanners, giving an improved approximation ratio of $O(n^{2/3+\epsilon})$. For DSF with *uniform* edge costs, Chlamtáč et al. obtained an $O(n^{3/5+\epsilon})$-approximation [19], which was subsequently improved to $O(n^{4/7+\epsilon})$ by Abboud and Bodwin [1].

Length-Constrained Network Design. Length-constrained network design has been primarily studied in undirected graphs. It was first introduced by Balakrishnan and Altinkemer [5], and has since been studied in both approximation and fast exact algorithms. The length-constrained s-t shortest path problem is solvable exactly in polynomial time when edge lengths are polynomially bounded, via the textbook Bellman-Ford dynamic programming algorithm. However, it is NP-hard for arbitrary lengths [42].[2] Moreover, length-constrained MST is NP-hard even in the special case of hop constraints.[3] From an approximation algorithms standpoint, until recently, the majority of the work on length-constrained network design had been limited to single-source settings, such as s-t shortest path [43, 59, 38, 32, 53, 8], MST [21, 2, 55, 3, 47, 56, 54, 40], Steiner Tree [49, 44], and k-Steiner Tree [36, 46]. While s-t shortest path admits an FPTAS [38], other length-constrained problems have strong hardness results under *strict* length constraints. Length-constrained MST is as hard as Directed Steiner Tree, and the length-constrained variant of Steiner forest has no $o(2^{\log^{1-\epsilon}(n)})$-approximation [23]. These problems are more tractable when length constraints are relaxed. A simple merging algorithm [54] yields an $(O(\log n), O(\log n))$-bicriteria approximation for length-constrained MST and Steiner Tree. Recent progress by Hershkowitz and Huang allows one to reduce the length slack to $O(\log n/\log\log n)$ while increasing the approximation ratio by a polylogarithmic factor [40]. The multicommodity setting has been more challenging. For length-constrained Steiner Forest, polylogarithmic bicriteria approximations were implicitly known via buy-at-bulk [11, 48]. There has also been a line of work on hop-constrained *tree embeddings*, which have led to bicriteria approximations for hop-constrained Steiner Forest and various related problems [35, 26]. Chekuri and Jain [13], building on hop-constrained oblivious routing tools of Ghaffari, Haeupler, and Zuzic [30], obtained LP-competitive algorithms for multicommodity problems in buy-at-bulk and length-constrained settings. In directed graphs, a recent work obtains polynomial factor approximations for Directed Buy-at-Bulk and related spanner problems [34]; these approximation ratios essentially match the best known factors for DSF in general digraphs.

Planar and Minor-Free Graphs. Improved approximation ratios have been obtained for several problems in special classes of graphs, such as planar and minor-free graphs. We first discuss undirected graphs. In planar graphs, Steiner Tree admits a PTAS [9]; this was later extended to a PTAS for Steiner Forest in graphs of bounded genus [6]. Recently, [20] obtained a QPTAS for Steiner Tree in minor-free graphs. Furthermore, although the node-weighted variant of Steiner Tree captures Set Cover in general graphs, there exists a constant factor approximation in planar graphs, and more generally, in any proper minor-closed graph family [22]. In directed graphs, several recent results have built on tools given by Thorup [58] to obtain approximation algorithms in planar digraphs. Kawarabayashi and Sidiropoulos [45] gave a polylogarithmic upper bound on the multicommodity flow-cut gap in planar digraphs; this was derandomized and extended to the node-weighted setting by [15]. As discussed earlier, Friggstad and Mousavi [29] obtained an $O(\log k)$-approximation for DST in planar digraphs. This was extended by Chekuri et al. [17] to an LP-competitive $O(\log^2 k)$-approximation. Chekuri et al. [17] also obtained polylogarithmic approximations for several rooted variants of DST in planar digraphs, such as group, covering, polymatroid, and multirooted versions. Chekuri and Jain [14] then obtained an $O(\log^6 k)$-approximation for DSF in planar digraphs. In minor-free input graphs that are *quasi-bipartite*, Friggstad and Mousavi [28] obtained a constant-factor approximation for DST.[4]

[2] The length-constrained shortest path problem is often referred to as *Restricted Shortest Path*.
[3] Hop constraints refer to the unit-edge-length setting of length-constrained network design.
[4] The input is quasi-bipartite if there are no edges between any two non-terminal nodes.

2 Preliminaries

2.1 Notation

Let $G = (V, E)$ be a directed graph with edge costs $c : E \to \mathbb{R}_+$. For $E' \subseteq E$, we denote $c(E') = \sum_{e \in E'} c(e)$. We assume all edge costs $c(e) \geq 1$ and are polynomially bounded in n. This can be done with a $(1 + o(1))$ factor loss in approximation, by guessing the cost of the optimal solution OPT, contracting edges with cost much smaller than OPT, and scaling appropriately.

For $S \subseteq V$, we use $E[S]$ to denote the set of edges of E with both endpoints in S, $G[S]$ to denote the subgraph $(S, E[S])$ *induced by* S in G, and $\delta^+(S) = \{(u, v) \in E : u \in S, v \notin S\}$ to denote the *out-cut* of S. The *height* of an out-tree T rooted at s is the maximum over all $v \in V(T)$ of the number of edges on the unique s-v path. The *size* of the tree is the number of vertices in the tree. To *contract* an edge (u, v) is to merge its endpoints into a single vertex w_{uv}, replacing all edges incident to u or v with edges incident to w_{uv} and removing resulting self-loops. For a subgraph $G' \subseteq G$, we use G/G' to denote the graph obtained by contracting every edge of G' and $G - G'$ to denote the graph obtained by deleting every edge in G'. A graph H obtained from G via edge contractions and edge or vertex deletions is called a *minor* of G. It is easy to see that if G is planar, then any minor of G is planar as well. A weakly connected component of G is a maximal connected component of the underlying undirected graph obtained from G by ignoring the edge orientations. We call this undirected graph the *undirected version* of G.

The length $\ell(p)$ of a path p is the sum of the lengths of its edges. More generally, we write $\ell(F) := \sum_{e \in F} \ell(e)$ for any $F \subseteq E$. For any subset of edges $F \subseteq E$ and any pair of vertices $u, v \in V$, we let $\ell_F(u, v)$ denote the length of the shortest-length u-v path in F. We generalize this notation to sets: for $A, B \subseteq V$, $\ell_F(A, B)$ is the length of the shortest-length path in F from A to B. We drop F when it is clear from context. For a tree T and vertices $u, v \in V(T)$, we write $P_T(u, v)$ to denote the unique u-v path in T.

2.2 Mixture Metric

We write $d^{(h)}(s, t)$ to denote the cost of the shortest h-length constrained s-t path. Computing $d^{(h)}(s, t)$ is NP-hard [42]; however, it can be approximated to a $(1 + \epsilon)$-factor [38]. Unlike traditional graph distances, $d^{(h)}$ does not form a metric. We instead consider the *mixture metric*; this is a standard notion to facilitate working with length constraints, and was highlighted in recent work in length-constrained network design by [35].

Given a length constraint h and a cost bound γ, we define the mixture metric as follows: for each $e \in E$, the *mixture cost* function is $c'(e) = c(e) + \frac{\gamma}{h} \ell(e)$. For all $s, t \in V$, let $d'(s, t)$ be the cost of the shortest s-t path with respect to c'. It is not difficult to see that d' forms a metric. Claim 6 demonstrates how d' approximates $d^{(h)}$; the proof is simple and deferred to a full version.

▷ Claim 6. Let d' be the mixture metric with length constraint h and cost bound γ.

1. $d'(s, t) \leq d^{(h)}(s, t) + \gamma$;
2. $d^{(h')}(s, t) \leq d'(s, t)$, where $h' = \frac{d'(s,t)}{\gamma} \cdot h$.

2.3 Planar Separator

The key property of planarity used in our algorithms is the fact that planar graphs have good shortest path separators, as shown in Lemma 5. We state a more general lemma on *undirected graphs*. This lemma, essentially proved by Lipton and Tarjan [52], but given explicitly by Thorup [58], states that every spanning tree on any planar graph contains three paths whose removal results in connected components each containing at most half the number of vertices. The formal statement is as follows.

▶ **Lemma 7** ([52, 58]). *Let G be an undirected connected graph with non-negative edge costs $c(e)$, non-negative vertex weights $w(u)$, and a spanning tree T rooted at $v \in V$. There exists a polynomial time algorithm to find three vertices $u_1, u_2, u_3 \in V$ such that each component of $G \setminus (P_T(v, u_1) \cup P_T(v, u_2) \cup P_T(v, u_3))$ has at most half the vertex weight of G.*

Note that one can recover the directed variant (Lemma 5) by considering the spanning tree T to be a shortest path tree rooted at v.

3 Length-Constrained Directed Steiner Tree

In this section, we consider length-constrained Directed Steiner Tree. We provide an $O(\log k)$-approximation in Section 3.1, and an LP-based $O(\log^2 k)$-approximation in Section 3.2.

First, we define a subroutine PruneAndSeparate (Algorithm 1) which will be used in both the integral and LP-based algorithms. This subroutine takes as input a graph $G = (V, E)$ with edge costs $c(e)$, a root $s \in V$, terminals $S \subseteq V$, and a guess γ for the cost of the optimal solution. PruneAndSeparate(G, s, S, γ) removes all vertices further than γ away from s, and uses Lemma 5 on the resulting graph with vertex weights set to 1 on the terminals and 0 elsewhere. This yields a *planar separator* $P := P_1 \cup P_2 \cup P_3$ in which each resulting component of $G \setminus P$ has at most half the terminals. The subroutine contracts P into s; each component of $G \setminus P$ corresponds to a new subinstance induced by the terminals in that component along with s. PruneAndSeparate(G, s, S, γ) returns P along with the subinstances for each component.

Algorithm 1 PruneAndSeparate(G, s, S, γ).

Delete all vertices $v \in V$ with $d_G(s, v) > \gamma$
Let $P := P_1 \cup P_2 \cup P_3$ be given by Lemma 5 with weights $w(v) = \mathbb{1}_{v \in S}$ for $v \in V(G)$
Let G_P be obtained from G by contracting P into s
Let $C_1, \ldots, C_\tau$ be components of $G \setminus P$, and let $G_i \leftarrow G_P[C_i \cup \{s\}]$
return $(P, (G_1, C_1), \ldots, (G_\tau, C_\tau))$

3.1 Approximation with respect to Integral Optimum

We follow the divide-and-conquer algorithm given by Friggstad and Mousavi [29], adapted to handle length constraints. Recall the high level approach: the algorithm branches on two cases and returns the cheaper solution. In the first branch, it recurses with a halved guess for OPT; this handles the case where OPT $\leq \gamma/2$. In the second branch, it computes a planar separator P under the *mixture cost* c', and recursively solves each resulting component independently. Using the mixture cost for the separator is the main adaptation for length constraints, which ensures that the separator paths have bounded cost *and* bounded length. For clarity, we write the algorithm as though $d^{(h)}(s, t)$ can be computed exactly; replacing this with a $(1 + \epsilon)$-approximation does not affect the approximation guarantees.

Algorithm 2 h-DST(I, γ): an instance $I = (G, s, S, h)$ and a "guess" γ for OPT.

if $\exists t \in S$ such that $d^{(h)}(s,t) > \gamma$ **then**
  return Infeasible (∞)
if $|S| = 1$ **then**
  return min-cost h-length-constrained s-t path in G, where t is the terminal in S
// **Recursive Step 1:**
$F^1 \leftarrow$ h-DST$(I, \gamma/2)$
// **Recursive Step 2:**
Define mixture cost $c'(e) = c(e) + \frac{\gamma}{h}\ell(e)$
$(P, (G_1, C_1), \ldots, (G_\tau, C_\tau)) \leftarrow$ PruneAndSeparate$(G = (V, E, c'), s, S, 2\gamma)$
$F^2 \leftarrow P$
for $i \in [\tau]$ **do**
  $F_i \leftarrow$ h-DST$(I_i := (G_i, s, S \cap C_i, h), \gamma)$
  $F^2 \leftarrow F^2 \cup F_i$
return $\arg\min_{F \in \{F^1, F^2\}} c(F)$

First, we argue that the algorithm runs in polynomial time. This follows directly from [29]; the proof is deferred to a full version.

▶ **Lemma 8.** *Algorithm 2 runs in polynomial time.*

Next, we prove feasibility and bound the cost of the given solution. The proof of Lemma 11 is deferred to a full version. We start with a simple property about the planar separator.

▷ Claim 9. The planar separator P computed by h-DST(I, γ) for any feasible instance satisfies the following: (i) $c(P) \leq 6\gamma$, (ii) $\ell(P) \leq 6h$.

Proof. Let $P = P_1 \cup P_2 \cup P_3$ be the planar separator computed by h-DST(I, γ). Recall that each P_i, for $i \in [3]$, is a shortest path from s with respect to c'. Since all vertices $u \in V$ with $d_{G,c'}(s, u) > 2\gamma$ are removed, each path P_i must have $c'(P_i) = c(P_i) + \frac{\gamma}{h}\ell(P_i) \leq 2\gamma$. Since both terms are non-negative, $c(P_i) \leq 2\gamma$, so $c(P) \leq 6\gamma$. Similarly, $\frac{\gamma}{h}\ell(P_i) \leq 2\gamma$, so $\ell(P_i) \leq 2h$ and thus $\ell(P) \leq 6h$. ◁

▶ **Lemma 10.** *Let F be the solution returned by* h-DST(I, γ) *for a feasible instance (I, γ). Then, F contains an s-t path of length at most $O(\log |S|) \cdot h$ for each terminal $t \in S$.*

Proof. We prove by induction on $\gamma + |S|$ that if the instance (I, γ) is feasible, then $F :=$ h-DST(I, γ) contains a path from s to t of length at most $6h(\log |S| + 1)$, for each $t \in S$. The base case is clear; if there is one terminal, then F is an s-t path of length h.

We consider the two recursive cases separately. For the first recursive case, if $(I, \gamma/2)$ is infeasible, then $c(F^1)$ is infinite and $F \neq F^1$. Else, by induction, F^1 contains an s-t path of length at most $6h(\log |S| + 1)$ for each $t \in S$.

Next, we consider the second recursive case. Fix $t \in S$. Observe that since (I, γ) is feasible, $d^{(h)}(s, t) \leq \gamma$, so there exists an s-t path q of length at most h and cost at most γ. In particular, $c'(q) = c(q) + \frac{\gamma}{h}\ell(q) \leq 2\gamma$, so t is not deleted in the pruning step. Therefore, t is either in the planar separator P or in some component C_i of $G \setminus P$. If $t \in P$, then P contains an s-t path of length at most $\ell(P) \leq 6h$, by Claim 9. If $t \in C_i$, then since the recursive sub-instance contracts P into s, C_i contains a path from $\{s\} \cup P$ to t of length at most $6h(\log |S \cap C_i| + 1)$, by induction. Therefore, F^2 contains an s-t path of length at most

$$\ell(P) + 6h(\log |S \cap C_i| + 1) \leq 6h + 6h(\log(|S|/2) + 1) = 6h + 6h \log |S| = 6h(\log |S| + 1).$$

This concludes the inductive argument and the proof of the lemma. ◀

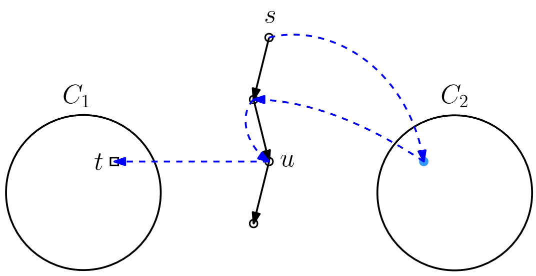


Figure 3 Consider the s-t path p shown in blue dashed lines; here the separator P is black and bolded. The portion of the path from u to t corresponds to p'.

▶ **Lemma 11.** *The cost of the solution returned by* h-DST(I,γ) *for a feasible instance* (I,γ) *is at most* $O(\log|S|)\cdot\mathrm{OPT}$.

Theorem 2 follows from Lemma 8, Lemma 10, and Lemma 11 by applying Algorithm 2 to the given input instance I, with γ chosen to be any upper bound on OPT; e.g. $c(E)$.

3.2 LP-Based Approximation

The goal of this section is to obtain a polylogarithmic upper bound on the integrality gap of a natural LP relaxation for LC-DST in planar digraphs, defined as follows. For each terminal $t \in S$, let $\mathcal{P}_t^h$ denote the set of all s-t paths in G of length at most h. The variables x_e indicate whether an edge e is in the final solution, and variables f_p indicate how much s-t flow is sent on path p.

$$\begin{aligned}\min \quad & \sum_{e\in E} c(e)x_e \\ s.t. \quad & \sum_{p\in\mathcal{P}_t^h} f_p \geq 1 && \forall t\in S \\ & \sum_{p\in\mathcal{P}_t^h, e\in p} f_p \leq x_e && \forall e\in E, t\in S \\ & x_e, f_p \geq 0 && \forall e\in E, p\in \cup_{t\in S}\mathcal{P}_t^h\end{aligned} \tag{LC-DST-LP}$$

The LP contains an exponential number of variables but can be solved approximately (up to a $(1+\epsilon)$ factor) via separation oracle on the dual.

We provide an LP-based algorithm following a similar framework as above; here, we use the LP solution as the "guess" for OPT. This is the same approach used by [17] to obtain an LP-based approximation for DST in planar graphs. The algorithm h-frac-DST$(I,(x,f))$ is as follows, where $I=(G,s,S,h)$ is a problem instance and (x,f) is a feasible LP solution to the relaxation for LC-DST-LP.

- **Base Case.** If $S=\{t\}$, we return the min-cost h-length constrained s-t path.
- **Prune and Separate.** We set $\gamma = 2\log|S|\sum_{e\in E} c(e)x_e$ to be the "guess" for OPT. As in Algorithm 2, we define the mixture edge cost function as $c'(e) = c(e)+\frac{\gamma}{h}\ell(e)$. Let $(P,(G_1,C_1),\ldots,(G_\tau,C_\tau))$ denote the output of PruneAndSeparate$(G=(V,E,c'),s,S,2\gamma)$.
- **Recursive Step.** For each component $i\in[\tau]$, we define the recursive sub-instance $I_i=(G_i,s,S\cap C_i,h)$. In order to construct a feasible LP solution for I_i, we first scale up the solution; let $(\bar{x},\bar{f}) = \left(1+\frac{1}{\log|S|}\right)(x,f)$. We set $x_e^i = \bar{x}_e$ for all $e\in G_i$.

For the path variables, fix a terminal $t \in S \cap C_i$. For each path $p \in \mathcal{P}_t^h$ (in the graph G *before* contracting P into s), let p' denote the corresponding contracted path with resulting cycles deleted. Note that $\ell(p') \le h$ since contraction cannot increase path length, so p' is a valid s-t path in the new sub-instance. We define $\mathcal{Q}(p')$ as the set of all paths in $\mathcal{P}_t^h$ that map to p' after contraction and cycle removal; see Figure 3. We set $f_{p'}^i = \sum_{p \in \mathcal{Q}(p')} \bar{f}_p$ for all such p' fully contained in G_i. We recursively compute $F_i = \text{h-frac-DST}(I_i, (x^i, f^i))$.

- Return $P \cup (\cup_i F_i)$

It is clear that this algorithm runs in polynomial time. We show that each recursive call is valid by showing that the constructed LP solutions are feasible for the subinstances.

▶ **Lemma 12.** *For each $i \in [\tau]$, (x^i, f^i) is a feasible LP solution on instance I_i.*

Proof. Fix $i \in [\tau]$. We show that (x^i, f^i) satisfies all LP constraints. Fix $t \in S \cap C_i$. To distinguish between the set of h-length-constrained s-t paths in the new instance I_i versus in the original input instance I, we write $\mathcal{P}_t^i$ versus $\mathcal{P}_t$; the length-constraint h is dropped from this notation for visual clarity.

We start by showing that $\sum_{p \in \mathcal{P}_t, c'(p) \le 2\gamma} \bar{f}_p \ge 1$; that is, scaling up by a $1 + 1/\log|S|$ factor compensates for vertices deleted during the pruning step. Consider a path $p \in \mathcal{P}_t$ with $c'(p) > 2\gamma$. Since $p \in \mathcal{P}_t$, $\ell(p) \le h$. Thus, $c(p) = c'(p) - \frac{\gamma}{h}\ell(p) > 2\gamma - \gamma = \gamma$. Observe that since (x, f) satisfies the LP constraints,

$$\sum_e c(e) x_e \ge \sum_e c(e) \sum_{p \in \mathcal{P}_t : e \in p} f_p = \sum_{p \in \mathcal{P}_t} f_p c(p) > \gamma \sum_{p \in \mathcal{P}_t : c(p) > \gamma} f_p.$$

In particular, $\sum_{p \in \mathcal{P}_t : c(p) > \gamma} f_p < \frac{1}{\gamma} \sum_e c(e) x_e = 1/(2\log|S|)$. Furthermore, since (x, f) is feasible for I, $\sum_{p \in \mathcal{P}_t} f_p \ge 1$. Therefore,

$$\sum_{p \in \mathcal{P}_t, c'(p) \le 2\gamma} \bar{f}_p \ge (1 + \frac{1}{\log|S|}) \sum_{p \in \mathcal{P}_t : c(p) \le \gamma} f_p \ge \left(1 + \frac{1}{\log|S|}\right)\left(1 - \frac{1}{2\log|S|}\right) \ge 1.$$

Next, we show that $\sum_{p \in \mathcal{P}_t^i} f_p^i = \sum_{p \in \mathcal{P}_t, c'(p) \le 2\gamma} \bar{f}_p$. Let $p \in \mathcal{P}_t$, and let p' denote the corresponding contracted path. Since $c'(p) \le 2\gamma$, no vertices of p are deleted during the pruning step. We want to show that p' is contained in G_i. Let v be the *last* vertex on p that is not in C_i; note that p must end in C_i since p is an s-t path and $t \in C_i$. Since the components $C_1, \ldots, C_\tau$ are weakly connected in $G \setminus P$, there must be no edges between them. Thus, $v \in P \cup \{s\}$. Therefore, when P is contracted into s, the path $p[s, v]$ corresponds to a walk starting and ending at the contracted supernode s. This implies that $p' = p[v, t]$, which in the contracted graph, is contained in G_i. Therefore, $p \in \mathcal{Q}(p')$ for some $p' \in \mathcal{P}_t^i$, so $\sum_{p \in \mathcal{P}_t, c'(p) \le 2\gamma} \bar{f}_p \le \sum_{p' \in \mathcal{P}_t^i} \sum_{p \in \mathcal{Q}(p')} \bar{f}_p = \sum_{p' \in \mathcal{P}_t^i} f_{p'}^i$, as desired. Thus, $\sum_{p \in \mathcal{P}_t^i} f_p^i \ge 1$, so the first set of LP constraints is satisfied.

Proving that the second set of LP constraints are satisfied is more straightforward. We want to show that for all edges $e \in E(G[C_i \cup \{s\}])$, $\sum_{p \in \mathcal{P}_t^i, e \in p} f_p^i \le x_e^i$.

$$\sum_{p' \in \mathcal{P}_t^i, e \in p'} f_{p'}^i = \sum_{p' \in \mathcal{P}_t^i, e \in p'} \sum_{p \in \mathcal{Q}(p')} \bar{f}_p \le \sum_{p \in \mathcal{P}_t, e \in p} \bar{f}_p,$$

since the sets $\mathcal{Q}(p')$ are disjoint. To conclude, since (x, f) is a feasible LP solution,

$$\sum_{p \in \mathcal{P}_t, e \in p} \bar{f}_p = (1 + \frac{1}{\log|S|}) \sum_{p \in \mathcal{P}_t, e \in p} f_p \le (1 + \frac{1}{\log|S|}) x_e = x_e^i.$$ ◀

The "prune and separate" portion of h-frac-DST is the same as in h-DST (Algorithm 2). Therefore, Claim 9 holds for h-frac-DST. It is not difficult to see that the resulting solution $F = \text{h-frac-DST}(I, (x, f))$ contains an s-t path of length at most $O(\log |S|) \cdot h$ for each terminal $t \in S$, using reasoning analogous to that of Lemma 10. We bound the approximation ratio in the following lemma, which concludes the proof of Theorem 3.

▶ **Lemma 13.** *If (x, f) is a feasible LP solution to the LC-DST instance I, then the cost of the solution returned by* $\text{h-frac-DST}(I, (x, f))$ *is at most $O(\log^2 k) \sum_e c(e)x_e$.*

Proof. Let F be the solution returned by $\text{h-frac-DST}(I, (x, f))$. We let $X = \sum_e c(e)x_e$. We prove by induction on the number of terminals that $c(F) \leq 12(\log |S| + 1)^2 X$.

In the base case, it is easy to see that the cost of the min-cost h-length-bounded s-t path is at most X. For the recursive step, for each $i \in [\tau]$, let X_i denote the value of the LP solution (x^i, f^i), and let F_i denote the output of $\text{h-frac-DST}(I_i, (x^i, f^i))$. By Claim 9, $c(P) \leq 6\gamma = 12X \log |S|$. By induction,

$$c(F) = c(P) + \sum_{i \in [\tau]} c(F_i) \leq 12X \log |S| + \sum_{i \in [\tau]} 12X_i(\log |S \cap C_i| + 1)^2.$$

Since the number of terminals is halved at each recursive step, $\log |S \cap C_i| \leq \log(|S|/2) \leq \log |S| - 1$. Furthermore, since $C_1, \ldots, C_\tau$ are disjoint, $\sum_{i \in [\tau]} X_i \leq (1 + \frac{1}{\log |S|})X$. Thus

$$\begin{aligned} c(F) &\leq 12 \log |S| X + \sum_{i \in [\tau]} 12(\log |S|)^2 X_i \leq 12 \log |S| X + 12(\log |S|)^2 (1 + \frac{1}{\log |S|})X \\ &= 12X(\log^2 |S| + 2 \log |S|) \leq 12X(\log |S| + 1)^2. \end{aligned}$$

This concludes the inductive argument; thus, $c(F) \leq O(\log^2 k) \sum_{e \in E} c(e)x_e$. ◀

4 Length-Constrained Directed Steiner Forest

In this section, we consider length-constrained Directed Steiner Forest in planar digraphs (called length-constrained planar DSF). We are given as input a directed graph $G = (V, E)$ with non-negative edge costs $c(e)$, lengths $\ell(e)$, length constraint h, and demand pairs $D = \{(s_i, t_i)\}_{i \in [k]}$. The goal is to find a min-cost subgraph $H \subseteq G$ such that for each $(s_i, t_i) \in D$, H contains an s_i-t_i path of length at most h. We employ the *junction tree* approach, discussed in Section 1. We formally define an h-length-constrained junction tree.

▶ **Definition 14.** *Given an instance (G, D, h) of LC-DSF, an h-*length-constrained junction tree *that covers terminal pairs $D_H \subseteq D$ is a subgraph $H \subseteq G$ with a root $v \in V(H)$ such that for each $(s_i, t_i) \in D_H$, H contains an s_i-v path and a v-t_i path, each of length at most h. The* density *of H is $c(H)/|D_H|$.*

We first show, in Section 4.1, that given any instance to planar LC-DSF, there exists a low-density length-constrained junction tree. In Section 4.2, we describe the approximation algorithm to find such a junction tree, and conclude the proof of Theorem 4.

4.1 Existence of a good junction tree

We prove the following theorem.

▶ **Theorem 15.** *Given an instance (G, D, h) of length-constrained planar DSF, there exists a $(3h)$-length-constrained junction tree of density at most $O(\log^2 k) \cdot \text{OPT}/k$.*

For planar-DSF *without* length-constraints, [14] showed that every instance of planar-DSF contains a low density junction tree. We outline the approach of [14], as we follow a similar framework. The proof proceeds by considering an optimum solution $E^* \subseteq E$ and finding several junction trees in E^* that are essentially disjoint and cover a large fraction of terminal pairs. The construction of these junction trees follows three steps. The first is a "layering" of a directed graph, where each layer has a tree-like structure, and all paths in the original graph are contained in at most two consecutive layers. This layering scheme was given by Thorup [58] and also used by Kawarabayashi and Sidiropoulos [45] to obtain improved upper bounds on the multicommodity flow-cut gap in directed planar graphs. The second step is a recursive approach based on the planar separator theorem (Lemma 7). The third step is to show that in one level of recursion, if many s_i-t_i paths pass through the separator, then there exist vertices on the separator that behave as roots of low-density junction trees.

The proof of Theorem 15 follows the same three-step strategy. First, we reduce to the case where the optimal solution is an *h-length-constrained* 3*-layered digraph* (Lemma 17).

▶ **Definition 16.** *An h-length-constrained* 3*-layered spanning tree T of a digraph G is a spanning tree of the* undirected version *of G, such that any root-to-leaf path in T is the concatenation of at most* 3 *dipaths in G, each of length at most h. An h-length-constrained* 3*-layered digraph is one that has an h-length-constrained* 3*-layered spanning tree. The root of the digraph is the root of the spanning tree.*

The argument uses a similar decomposition as in [14], but requires additional care to handle lengths. Second, we apply the planar separator and corresponding divide-and-conquer argument to reduce to a so-called "one-path" setting (Lemma 20). This step is essentially identical to the corresponding argument in [14]. The third step is a new contribution: handling the one-path setting under length constraints. The argument of [14] no longer applies, and we give a new argument (Lemma 23).

4.1.1 Reduction to length-constrained layered digraphs

First, we reduce to the case where the optimal solution is an h-length-constrained 3-layered digraph, by proving the following lemma.

▶ **Lemma 17.** *Let (G, D, h) be an instance of length-constrained planar DSF and let $G^* = (V^*, E^*)$ be an optimal solution. Then, there exists a subgraph $H \subseteq G^*$ and a subset of terminals $D_H \subseteq D$ such that the following holds.*

1. *H is an h-length-constrained* 3*-layered digraph with root v;*
2. *For each $(s_i, t_i) \in D_H$, there exists an s_i-t_i path in $H \setminus \{v\}$ of length at most h;*
3. *H is a minor of G^*, and $H \setminus \{v\}$ is a subgraph of G^*.*
4. $\frac{c(E(H))}{|D_H|} \leq 3 \cdot \frac{c(E^*)}{|D|}$.

To prove Lemma 17, we will decompose G^* into a series of 3-layered digraphs and show that at least one satisfies the necessary properties. We assume G^* is weakly connected; else, this decomposition can be applied separately on each weakly connected component. Let v_0 be an arbitrary node in V^*. Let L_0 be the set of all nodes in V^* that are reachable from v_0 (in G^*) with a path of length at most h. We then define alternating layers until all vertices have been covered by a layer.

$$L_j = \begin{cases} \{u \in V^* \setminus \cup_{j'<j} L_{j'} : \ell(u, L_{j-1}) \leq h\} & j \text{ is odd} \\ \{u \in V^* \setminus \cup_{j'<j} L_{j'} : \ell(L_{j-1}, u) \leq h\} & j \text{ is even} \end{cases}$$

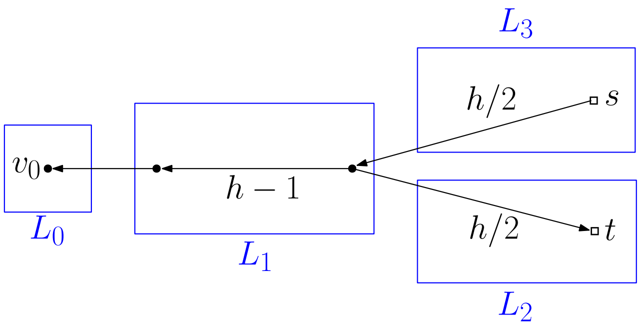


Figure 4 Example of graph in which s-t path spans 3 layers, L_1, L_2, L_3.

Let q denote the total number of layers. For each layer $j \leq q-2$, we define G_j to be the graph obtained from G^* by contracting $\cup_{j'<j} L_{j'}$ into a new node v_j, and deleting $\cup_{j''>j+2} L_{j''}$.

▷ Claim 18. For each demand pair $(s_i, t_i) \in D$, there exists $j \in \{0, \ldots, q-2\}$ such that $G_j \setminus \{v_j\}$ contains an s_i-t_i path of length at most h.

Proof. Let P_i be an s_i-t_i path of length at most h in G^*; such a path must exist as G^* is a feasible solution to the given LC-DSF instance. Let j be the minimum index such that $P_i \cap L_j \neq \emptyset$. We assume j is odd; the case where j is even is similar. Let u be in $P_i \cap L_j$. Then, since $\ell(u, t_i) \leq h$, t_i must be contained in $\cup_{j' \leq j+1} L_{j'}$. Since all of P_i can reach t_i with a path of length at most h, $P_i \subseteq \cup_{j' \leq j+2} L_{j'}$. Therefore, since L_j is the first layer that intersects with P_i, $P_i \subseteq L_j \cup L_{j+1} \cup L_{j+2} \subseteq G_j \setminus \{v_j\}$.[5] ◁

▶ Remark 19. When considering planar-DSF *without* length constraints, [14] shows that all s_i-t_i paths are contained in at most 2 layers. With length constraints, however, 3 layers are needed; see Figure 4 for an example of a graph in which the s-t path spans 3 layers.

Proof of Lemma 17. For each $j \in \{0, \ldots, q-2\}$, let D_j denote the set of terminal pairs for which $G_j \setminus v_j$ contains an s_i-t_i path of length at most h. We show that the first three properties required by Lemma 17 hold for all (G_j, D_j). The first and third properties follow by construction: each G_j is an h-length-constrained 3-layered digraph with root v_j, and G_j is constructed from G^* via contraction (to form the root) and deletion. The second property follows by definition of D_j.

To conclude, we show that there exists some (G_j, D_j) that satisfies the fourth property. By Claim 18, $\cup_j D_j = D$. Furthermore, it is not difficult to see that $\sum_{j<q-1} c(G_j) \leq 3c(E^*)$, since each edge of E^* is either contained in one layer L_j, or goes between two layers L_j and L_{j+1} – in either case, it exists only in G_j, G_{j-1}, and G_{j-2}. By an averaging argument, there must be some $j \in [q-2]$ such that

$$\frac{c(G_j)}{|D_j|} \leq \frac{\sum_j c(G_j)}{\sum_j |D_j|} \leq \frac{3c(E^*)}{|D|}. \qquad ◀$$

[5] We point out a minor technicality – this claim is not true when $j = 0$. However, one can circumvent this issue by creating a new dummy vertex as the root of G_0 and adding a 0-length edge to v_0.

4.1.2 Divide-and-conquer via planar separator

Next, we use the fact that planar digraphs have good separators to reduce to the so-called "one-path" case. We prove the following lemma.

▶ **Lemma 20.** *Let $G^* = (V^*, E^*)$ be a planar h-length-constrained 3-layered digraph with root v, and let $D \subseteq V^* \times V^*$ be a set of demand pairs such that for each $(s_i, t_i) \in D$, $G^* \setminus \{v\}$ has an s_i-t_i path of length at most h. Then, there exists a subgraph $H \subseteq G^* \setminus \{v\}$, an h-length path $P \subseteq H$, and subset of terminals $D_H \subseteq D$ satisfying the following.*

1. *For each $(s_i, t_i) \in D_H$, there exists an h-length s_i-t_i path in H that intersects P;*
2. $\frac{c(E(H))}{|D_H|} \leq O(\log |D|) \frac{c(E^*)}{|D|}$.

The proof follows the same argument as that of [14]; we rewrite it here for completeness and borrow the terminology. Fix an instance (G, D) of planar-DSF and a feasible solution E^* satisfying the conditions outlined in the statement of Lemma 20 above. Let $T^* \subseteq E^*$ be an h-length-constrained 3-layered spanning tree of G^*. We will follow a recursive process to partition D into subsets on which we build junction trees.

- Set vertex weights as $w(s_i) = w(t_i) = 1$ for all $(s_i, t_i) \in D$, and 0 otherwise.
- Consider the undirected version of E^* and apply Lemma 7 on the undirected version of spanning tree T^* with vertex weights w. From this, we obtain u_1, u_2, u_3.
- Since T^* is an h-length-constrained 3-layered tree, each (undirected) path $P_{T^*}(v, u_i)$ consists of at most 3 dipaths, each of length at most h. Remove the root v and let Q_i^j for $i, j \leq 3$ denote the 3 dipaths of $P_{T^*}(v, u_i) \setminus \{v\}$. Let $S_0 = \cup_{i,j \leq 3} Q_i^j$ denote this collection of at most 9 dipaths, we call this the *separator*.
- Define $D_0 \subseteq D$ to be the set of all terminal pairs (s_i, t_i) such that E^* contains an h-length s_i-t_i path that intersects one of the dipaths in S_0. Equivalently, $(s_i, t_i) \in D_0$ iff there exists an h-length s_i-t_i path $P_i \subseteq E^*$ such that $V(P_i) \cap V(S_0) \neq \emptyset$. We say these terminal pairs are *separated.*
- Let $\mathcal{C}_1$ be the set of weakly connected components of $G \setminus (\cup_{i \in [3]} P_{T^*}(v, u_i))$; we drop "weakly connected" and simply refer to these as "components" in the remainder of this section. Note that each $C \in \mathcal{C}_1$ has at most half the total number of terminals.

We then follow a recursive process, similar to PruneAndSeparate described in Section 3. For each $C \in \mathcal{C}_1$, we contract S_0 into v and recurse on the sub-instance consisting of $G^*[C \cup \{v\}]$ with terminal set $\{(s_i, t_i) \in D : s_i, t_i \in C\} \setminus D_0$. It is easy to verify that $G^*[C \cup \{v\}]$ is an h-length-constrained 3-layered digraph with root v. Furthermore, each (s_i, t_i) pair in the new terminal set has an h-length-bounded path that does not intersect v; else, (s_i, t_i) would have been included in D_0. Therefore, we can apply the same process as above, and repeat this recursive process until each component has at most one terminal. Since the number of terminals halves at each step, the recursion depth is at most $\lceil \log 2k \rceil = \lceil \log k \rceil + 1$.

We use the following notation. For an instance defined on component C at recursive depth j, we let S_j^C, D_j^C denote the separator and corresponding separated terminal pairs. We let $\mathcal{C}_j$ denote the set of all components at recursive depth j (in particular, $\mathcal{C}_0 = \{G^*\}$), and denote by $S_j := \cup_{C \in \mathcal{C}_j} S_j^C$ and $D_j := \cup_{C \in \mathcal{C}_j} D_j^C$. The following claim is simple; the proof is deferred to a full version.

▷ Claim 21. $D \subseteq \cup_{j=0}^{\lceil \log k \rceil + 1} D_j$.

▶ **Corollary 22.** *There exists a level of recursion $j^* \in \{0, \ldots, \lceil \log k \rceil + 1\}$ such that $|D_{j^*}| \geq \frac{k}{\lceil \log k \rceil + 2}$.*

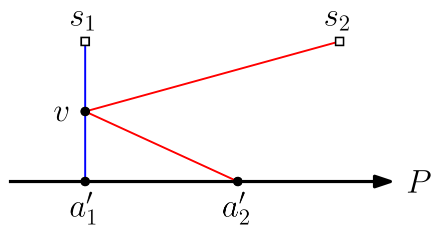


Figure 5 Suppose $\ell(s_1, v) = 1, \ell(v, a'_1) = h - 1, \ell(s_2, v) = h - 1, \ell(v, a'_2) = 1$. The argument in [14] would re-route s_2 to P via the path from v to a'_1; however, this would increase the length of the path from s_2 to P to $2h - 2$. In general, merging arbitrarily many terminals would make these paths to P arbitrarily long.

Proof of Lemma 20. Let j^* be the recursion level given by Corollary 22 such that $|D_{j^*}| \geq \frac{k}{\lceil \log k \rceil + 2}$. Since components of $\mathcal{C}_{j^*}$ are disjoint, $\sum_{C \in \mathcal{C}_{j^*}} c(E(C)) \leq c(E^*)$ and $\{D^C_{j^*}\}_{C \in \mathcal{C}_{j^*}}$ form a partition of D_{j^*}. Using an averaging argument, there exists a component C such that $c(E(C))/|D^C_{j^*}| \leq c(E^*)/|D_{j^*}|$. Furthermore, since $S^C_{j^*}$ consists of at most 9 dipaths, there exists at least one dipath Q^* such that at least $\frac{1}{9}$ terminal pairs of $D^C_{j^*}$ intersect Q^*; we denote this set of terminals by D^*.

We claim C, Q^*, D^* satisfy the required properties of the lemma statement. The first follows by construction; for all terminal pairs $(s_i, t_i) \in D^*$, there exists an h-length s_i-t_i path in C that intersects Q^*. For the second, $\frac{c(E(C))}{|D^*|} \leq 9 \cdot \frac{c(E(C))}{|D^C_{j^*}|} \leq 9 \cdot \frac{c(E^*)}{|D_{j^*}|} \leq O(\log k) \frac{c(E^*)}{|D|}$. ◀

4.1.3 One Path Case

To conclude, we show that in the "one-path" setting, there always exists a low-density junction tree. This subsection contains the core technical contribution over [14], since previously used techniques for handling the one-path setting do not extend to length-constraints.

▶ **Lemma 23.** *Let $G^* = (V^*, E^*)$ be a planar subgraph that contains an h-length path $P \subseteq G^*$, and let $D \subseteq V^* \times V^*$ be a set of demand pairs such that for each $(s_i, t_i) \in D$, there exists an h-length s_i-t_i path in G^* that intersects P. Then, G^* contains a $3h$-length-constrained junction tree of density $O(\log k) c(E^*)/k$, where $k = |D|$.*

Let $v_1, \ldots, v_{|P|}$ denote the vertices of P. For each terminal pair $(s_i, t_i) \in D$, the lemma assumption guarantees an h-length s_i-t_i path in G^* that intersects P. Let a'_i denote the first vertex of P on this path, and let b'_i denote the last vertex of P on this path; note that $a'_i \leq_P b'_i$. Let P_{s_i} be the subpath from s_i to a'_i, and let P_{t_i} be the subpath from b'_i to t_i; both have length at most h. We can assume without loss of generality that $E^* = P \cup (\cup_{i \in [k]} (P_{s_i} \cup P_{t_i}))$, since this satisfies the lemma assumptions. The main challenge with length constraints in the one-path setting is in handling edges of $E^* \setminus P$. In particular, the argument in [14] relies on the fact that for any $i, i' \in [k]$, if $P_{s_i} \cap P_{s_{i'}} \neq \emptyset$, then $a'_i = a'_{i'}$. This no longer holds, as demonstrated by Figure 5.

We adopt a different perspective. For each $(s_i, t_i) \in D$, let a_i denote the first node in P that s_i can reach within a path of length *at most* $2h$, and let b_i denote the last node in P that can reach t_i within length at most $2h$. We use the following simple claim.

▷ Claim 24. For all $i \in [k]$, $a_i \leq_P a'_i \leq_P b'_i \leq_P b_i$.

Proof. By definition, $a'_i \leq_P b'_i$. For the other two inequalities, $a_i \leq_P a'_i$, since a_i is the *first* node on P that s_i can reach within length $2h$, and P_{s_i} is an s-P path of length h. Similarly, $b'_i \leq_P b_i$, since b_i is the *last* node on P that can reach t_i within length $2h$, and P_{t_i} is a path from P to t_i of length h. ◁

Let μ be the midpoint of the path P (i.e., $\mu = v_{\lfloor |P|/2 \rfloor}$). We partition D into three sets:

$$\begin{cases} L = \{(s_i, t_i) : a_i \leq_P b_i <_P \mu\} \\ R = \{(s_i, t_i) : b_i \geq_P a_i >_P \mu\} \\ M = \{(s_i, t_i) : a_i \leq_P \mu \leq_P b_i\} \end{cases}$$

▷ Claim 25. E^* contains a $3h$-length-constrained junction tree with root μ on terminals M.

Proof. It suffices to show that for each $(s_i, t_i) \in M$, there exist $3h$-length paths in E^* from s_i to μ and from μ to t_i. By definition, $\ell(s_i, a_i) \leq 2h$. Furthermore, since $a_i \leq_P \mu$ and P is an h-length path, $\ell(a_i, \mu) \leq h$. Therefore, $\ell(s_i, \mu) \leq 3h$. The argument for t_i is similar. ◁

We follow a recursive process. First, we construct the junction tree H_μ with root μ given by Claim 25. We conflate indices with vertices, and let $(\mu - 1)$ and $(\mu + 1)$ denote the vertices directly before and after μ on P. At a high level, the idea is then to recurse on the sub-paths $P[0, \mu - 1]$, $P[\mu + 1, v_{|P|}]$, with corresponding paths in $E \setminus P$ to connect terminals. Formally, let $E_L = P[0, \mu - 1] \cup \big(\cup_{(s_i,t_i) \in L} (P_{s_i} \cup P_{t_i})\big)$, and $E_R = P[\mu + 1, v_{|P|}] \cup \big(\cup_{(s_i,t_i) \in R} (P_{s_i} \cup P_{t_i})\big)$. We recurse on the edge-induced subgraph $G_L := G[E_L]$ with terminal set L, and $G_R := G[E_R]$ with terminal set R. The following claim follows directly from Claim 24, and proves that these are valid recursive calls.

▷ Claim 26. For each $(s_i, t_i) \in D$, recall that a'_i, b'_i denote the endpoints on P of P_{s_i}, P_{t_i} respectively. If $(s_i, t_i) \in L$, then $a'_i, b'_i \in P[0, \mu - 1]$, and if $(s_i, t_i) \in R$, then $a'_i, b'_i \in P[\mu + 1, v_{|P|}]$.

We continue this recursive process until P consists of one vertex v, in which case all remaining terminal pairs can be routed through v. This process terminates after $O(\log |P|) = O(\log k)$ recursive steps, since $|P|$ is halved at each step. The following key lemma bounds the total cost of the junction trees constructed. Let $E_s = E(\cup_{i \in [k]} P_{s_i})$ and $E_t = E(\cup_{i \in [k]} P_{t_i})$.

▷ Claim 27. $E_s \cap E_L$ is disjoint from $E_s \cap E_R$. Similarly, $E_t \cap E_L$ is disjoint from $E_t \cap E_R$.

Proof. We prove the claim for E_s; the argument for E_t is similar. Suppose for the sake of contradiction there exists an edge $e = (u, v)$ in $E_s \cap E_L \cap E_R$. Since $e \in E_L \cap E_s$, there must be a terminal s_i such that $e \in P_{s_i}$ and $a_i \leq_P b_i <_P \mu$. By Claim 24, $a'_i <_P \mu$. Similarly, since $e \in E_R$, there exists s_j such that $e \in P_{s_j}$ and $a_j >_P \mu$. Consider the path $P' = P_{s_j}[s_j, u] \circ P_{s_i}[u, a'_i]$. Since P_{s_j} and P_{s_i} are both h-length paths, P' has length at most $2h$. Thus, s_j can reach a'_i with a path of length $\leq 2h$, and $a'_i <_P a_j$, a contradiction. ◁

Proof of Lemma 23. The recursive process described yields a series of junction trees denoted H_v for some vertices $v \in P$ that, in total, cover all terminal pairs. By Claim 25, each H_v is a $3h$-length-constrained junction tree. To bound the cost, it is helpful to think of E_s, E_t and $E(P)$ as disjoint sets; this can be achieved by making at most 3 copies of each edge in E^*. Then, by Claim 27, E_L is disjoint from E_R. Therefore, each edge is in at most one junction tree per recursive level, so it is in at most $O(\log k)$ total junction trees. Therefore, the total cost of all junction trees is at most $3 \cdot O(\log k) c(E^*) = O(\log k) c(E^*)$. By an averaging argument, there must be one junction tree of density at most $O(\log k) c(E^*)/k$. ◀

To conclude, the proof of Theorem 15 follows from Lemma 17, Lemma 20, and Lemma 23.

4.2 Finding an Approximate Low-Density Junction Tree

The main lemma we prove in this section is the following.

▶ **Lemma 28.** *Suppose there exists a bicriteria (α, β)-approximation for LC-DST in planar graphs with respect to the optimal solution to LC-DST-LP. Then, given a planar-LC-DSF instance (G, D, h), there exists an efficient algorithm to obtain a βh-length-constrained junction tree of G of density at most $O(\alpha \cdot \log k)$ times the optimal.*

The algorithm is as follows. First, we guess the root of the junction tree; we can run this algorithm for each candidate root $v \in V$ and return the junction tree with minimum density. Next, we solve the following LP relaxation for the min-density junction tree rooted at v. We follow a similar structure to that of LC-DST-LP, with additional variables y_i for each $i \in [k]$ to indicate whether or not the pair (s_i, t_i) is covered in the solution. For each $(s_i, t_i) \in D$, we let $\mathcal{P}^h_{s_i}$ denote the set of all h-length-constrained paths from s_i to v, and $\mathcal{P}^h_{t_i}$ denote the set of all h-length-constrained paths from v to t_i. Following LC-DST-LP, we have variables x_e for each $e \in E$ indicating whether an edge e is in junction tree, and variables f_p indicating which paths are used to connect s_i to v and v to t_i. The resulting density of the junction tree would be $(\sum_{e\in E} c(e)x_e)/(\sum_{i\in[k]} y_i)$; we normalize $\sum_{i\in[k]} y_i = 1$.

$$
\begin{aligned}
\min \quad & \sum_{e\in E} c(e)x_e \\
s.t. \quad & \sum_{p\in\mathcal{P}^h_{s_i}} f_p \geq y_i && \forall i \in [k] \\
& \sum_{p\in\mathcal{P}^h_{t_i}} f_p \geq y_i && \forall i \in [k] \\
& \sum_{p\in\mathcal{P}^h_{s_i}, e\in p} f_p \leq x_e && \forall e \in E, i \in [k] \\
& \sum_{p\in\mathcal{P}^h_{t_i}, e\in p} f_p \leq x_e && \forall e \in E, i \in [k] \\
& \sum_{i\in[k]} y_i = 1 \\
& x_e, f_p, y_i \geq 0 && \forall e \in E, p \in \cup_{i\in[k]} \left(\mathcal{P}^h_{s_i} \cup \mathcal{P}^h_{t_i}\right), i \in [k]
\end{aligned}
\tag{LC-Den-LP}
$$

Let x^*, f^*, y^* be an optimal solution to LC-Den-LP. Partition the set of terminal pairs into groups $D_0, \dots, D_{\log k}$, where $(s_i, t_i) \in D_j$ if $y^*_i \in (\frac{1}{2^{j+1}}, \frac{1}{2^j}]$. We will abuse notation and sometimes write $i \in D_j$ for ease.

▷ Claim 29. There exists $\theta \in \{0, \dots, \log k\}$ such that $\sum_{i\in D_\theta} y^*_i \geq 1/(2(\log k + 1))$.

Proof. If $(s_i, t_i) \in D \setminus (\cup_{j=0}^{\log k} D_j)$, then $y^*_i \leq \frac{1}{2^{\log k+1}} = \frac{1}{2k}$. Thus, $\sum_{i\notin\cup_{j=0}^{\log k} D_j} y^*_i \leq \sum_{i\notin\cup_{j=0}^{\log k} D_j} \frac{1}{2k} \leq \frac{1}{2}$. Recall that $\sum_{i\in[k]} y^*_i = 1$. Therefore, $\sum_{i\in\cup_{j=0}^{\log k} D_j} y^*_i \geq \frac{1}{2}$. Since there are $\log k + 1$ disjoint groups, there must be at least one group D_θ with $\sum_{i\in D_\theta} y^*_i$ at least $1/(2(\log k + 1))$. ◁

Let θ be given by Claim 29. We use the (α, β)-approximation algorithm for LC-DST twice: first, we consider the instance on G with root v and terminal set $D^t_\theta = \{t_i : (s_i, t_i) \in D_\theta\}$, and obtain a solution T_t. Second, we let G^R be obtained from G by reversing the direction of all edges. We consider the LC-DST instance on G^R with root v and terminal set

$D_\theta^s = \{s_i : (s_i, t_i) \in D_\theta\}$ and obtain a solution T_s' in G^R. Note that T_s' contains an h-length-constrained v-s_i path in G^R for all $s_i \in D_\theta^s$. Thus, if we reverse all edges of T_s', the resulting graph T_s is a subgraph of G containing an h-length-constrained s_i-v path for all $s_i \in D_\theta^s$. Therefore, $T = T_t \cup T_s$ is a (βh)-length-constrained junction tree on terminal pairs D_θ.

▷ Claim 30. Let $\lambda = 2^{\theta+1}$. Then $(\lambda x^*, \lambda f^*)$ is a feasible solution to LC-DST-LP on both of the following instances:

- G with terminal set D_θ^t,
- G^R with terminal set D_θ^s.

In both cases, f^* is restricted to relevant paths for terminals in D_θ^t, D_θ^s.

Proof. We prove the claim for G^R with terminal set D_θ^s, as the argument for (G, D_θ^t) is analogous. Recall that $\mathcal{P}_{s_i}^h$ is the set of all h-length-constrained s_i-v paths in G. Thus, each path in $\mathcal{P}_{s_i}^h$ is an h-length constrained v-s_i path in G^R. Since (x^*, f^*, y^*) is a feasible solution to LC-Den-LP, $\sum_{p\in\mathcal{P}_{s_i}^h} f_p^* \geq y_i^*$. Furthermore, since $i \in D_\theta$, $y_i^* > \frac{1}{2^{\theta+1}}$. Thus, $\sum_{p\in\mathcal{P}_{s_i}^h} \lambda f_p^* \geq \lambda y_i^* > 1$.

The fact that $(\lambda x^*, \lambda f^*)$ satisfies the second constraint of LC-DST-LP follows immediately from the fact that (x^*, f^*) satisfies the third constraint of LC-DST-LP, since both x^* and f^* are scaled by the same value. ◁

▷ Claim 31. The density of T is at most $O(\alpha \log k) \sum_{e\in E} c(e)x_e^*$.

Proof. The algorithms used to construct T_s and T_t are (α, β)-approximations with respect to the LP. This, along with Claim 30, implies that the costs of T_t and T_s are each upper bounded by $\alpha\lambda \sum_{e\in E} c(e)x_e^*$, where $\lambda = 2^{\theta+1}$.

To bound $|D_\theta|$, observe that $\sum_{i\in D_\theta} y_i^* \leq \sum_{i\in D_\theta} 1/2^\theta = |D_\theta|/2^\theta$. Furthermore, recall that θ was defined by Claim 29 such that $\sum_{i\in D_\theta} y_i^* \geq 1/(2(\log k + 1))$. Therefore, $|D_\theta| \geq 2^\theta \sum_{i\in D_\theta} y_i^* \geq \frac{2^\theta}{2(\log k+1)}$. The density of T is at most

$$\frac{c(T_t) + c(T_s)}{|D_\theta|} \leq \frac{2\log k + 2}{2^\theta}\left(2\alpha\lambda \sum_{e\in E} c(e)x_e^*\right) = 8\alpha(\log k + 1)\sum_{e\in E} c(e)x_e^*. \quad ◁$$

Lemma 28 follows from Claim 31 and the fact that T is a (βh)-length-constrained junction tree. Combining Lemma 28 with the bicriteria LP-competitive approximation for LC-DST in planar digraphs (Theorem 3) then yields Theorem 32.

▶ **Theorem 32.** *Given an instance (G, D, h) of LC-DSF in planar digraphs, there exists an efficient algorithm to obtain an $O(\log k)$-length-constrained junction tree of G of density at most $O(\log^3 k)$ times the optimal.*

This, along with Theorem 15, yields an efficient algorithm to find an $O(\log k)$-length-constrained junction tree of density at most $O(\log^5 k)\mathrm{OPT}/k$. By the standard covering argument, this yields a bicriteria $(O(\log^6 k), O(\log k))$-approximation for LC-DSF in planar digraphs, concluding the proof of Theorem 4.

References

1. Amir Abboud and Greg Bodwin. Reachability preservers: New extremal bounds and approximation algorithms. *SIAM Journal on Computing*, 53(2):221–246, 2024. doi:10.1137/21M1442176.
2. Ibrahim Akgün and Barbaros Ç Tansel. New formulations of the hop-constrained minimum spanning tree problem via Miller-Tucker-Zemlin constraints. *European Journal of Operational Research*, 212(2):263–276, 2011. doi:10.1016/J.EJOR.2011.01.051.

3 Ernst Althaus, Stefan Funke, Sariel Har-Peled, Jochen Könemann, Edgar A. Ramos, and Martin Skutella. Approximating k-hop minimum-spanning trees. *Operations Research Letters*, 33(2):115–120, 2005. doi:10.1016/j.orl.2004.05.005.

4 Spyridon Antonakopoulos, Chandra Chekuri, Bruce Shepherd, and Lisa Zhang. Buy-at-bulk network design with protection. *Mathematics of Operations Research*, 36(1):71–87, 2011. doi:10.1287/moor.1110.0484.

5 Anantaram Balakrishnan and Kemal Altinkemer. Using a hop-constrained model to generate alternative communication network design. *ORSA Journal on Computing*, 4(2):192–205, 1992. doi:10.1287/ijoc.4.2.192.

6 M. Bateni, M. Hajiaghayi, and D. Marx. Approximation schemes for steiner forest on planar graphs and graphs of bounded treewidth. *Journal of the ACM (JACM)*, 58(5):21, 2011.

7 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and directed Steiner forest. *Information and Computation*, 222:93–107, 2013. 38th International Colloquium on Automata, Languages and Programming (ICALP 2011). doi:10.1016/j.ic.2012.10.007.

8 Aaron Bernstein. Near linear time $(1+\epsilon)$-approximation for restricted shortest paths in undirected graphs. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 189–201. SIAM, 2012. doi:10.1137/1.9781611973099.17.

9 Glencora Borradaile, Philip Klein, and Claire Mathieu. An $O(n \log n)$ approximation scheme for Steiner tree in planar graphs. *ACM Transactions on Algorithms (TALG)*, 5(3):1–31, 2009. doi:10.1145/1541885.1541892.

10 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. doi:10.1006/JAGM.1999.1042.

11 C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximation algorithms for nonuniform buy-at-bulk network design. *SIAM Journal on Computing*, 39(5):1772–1798, 2010. doi:10.1137/090750317.

12 Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed steiner network problem. *ACM Transactions on Algorithms (TALG)*, 7(2):1–17, 2011. doi:10.1145/1921659.1921664.

13 Chandra Chekuri and Rhea Jain. Approximation algorithms for hop constrained and buy-at-bulk network design via hop constrained oblivious routing. In Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman, editors, *32nd Annual European Symposium on Algorithms (ESA 2024)*, volume 308 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:21, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2024.41.

14 Chandra Chekuri and Rhea Jain. A polylogarithmic approximation for Directed Steiner Forest in planar digraphs. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2095–2110, 2025. doi:10.1137/1.9781611978322.67.

15 Chandra Chekuri and Rhea Jain. Node-weighted multicut in planar digraphs. *arXiv preprint*, 2026. doi:10.48550/arXiv.2601.20038.

16 Chandra Chekuri and Rhea Jain. A polylogarithmic approximation for buy-at-bulk network design with protection. In *Proceedings of the Fifty-Eighth Annual ACM Symposium on Theory of Computing (STOC)*, 2026. to be published.

17 Chandra Chekuri, Rhea Jain, Shubhang Kulkarni, Da Wei Zheng, and Weihao Zhu. From Directed Steiner Tree to Directed Polymatroid Steiner Tree in Planar Graphs. In Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman, editors, *32nd Annual European Symposium on Algorithms (ESA 2024)*, volume 308 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:19, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2024.42.

18 Chandra Chekuri and Martin Pal. A recursive greedy algorithm for walks in directed graphs. In *46th annual IEEE symposium on foundations of computer science (FOCS'05)*, pages 245–253. IEEE, 2005.

19 Eden Chlamtáč, Michael Dinitz, Guy Kortsarz, and Bundit Laekhanukit. Approximating spanners and directed steiner forest: Upper and lower bounds. *ACM Transactions on Algorithms (TALG)*, 16(3):1–31, 2020. doi:10.1145/3381451.

20 Vincent Cohen-Addad. Bypassing the surface embedding: approximation schemes for network design in minor-free graphs. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 343–356, New York, NY, USA, June 2022. Association for Computing Machinery. doi:10.1145/3519935.3520049.

21 Geir Dahl. The 2-hop spanning tree problem. *Operations Research Letters*, 23(1):21–26, 1998. doi:10.1016/S0167-6377(98)00029-7.

22 Erik D Demaine, MohammadTaghi Hajiaghayi, and Philip N Klein. Node-weighted Steiner tree and group Steiner tree in planar graphs. *ACM Transactions on Algorithms (TALG)*, 10(3):1–20, 2014. doi:10.1145/2601070.

23 Michael Dinitz, Guy Kortsarz, and Ran Raz. Label cover instances with large girth and the hardness of approximating basic k-spanner. *ACM Trans. Algorithms*, 12(2), December 2016. doi:10.1145/2818375.

24 Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *Proceedings of the thirty-first annual ACM Symposium on Theory of Computing (STOC)*, pages 750–759, 1999. doi:10.1145/301250.301447.

25 Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved approximation algorithms for Directed Steiner Forest. *Journal of Computer and System Sciences*, 78(1):279–292, January 2012. doi:10.1016/j.jcss.2011.05.009.

26 A. Filtser. Hop-constrained metric embeddings and their applications. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 492–503, Los Alamitos, CA, USA, February 2022. IEEE Computer Society. doi:10.1109/FOCS52979.2021.00056.

27 Zachary Friggstad, Jochen Könemann, Young Kun-Ko, Anand Louis, Mohammad Shadravan, and Madhur Tulsiani. Linear programming hierarchies suffice for directed Steiner tree. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 285–296. Springer, 2014. doi:10.1007/978-3-319-07557-0_24.

28 Zachary Friggstad and Ramin Mousavi. A constant-factor approximation for quasi-bipartite directed Steiner tree on minor-free graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, pages 13:1–13:18, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.13.

29 Zachary Friggstad and Ramin Mousavi. A $O(\log k)$-Approximation for Directed Steiner Tree in Planar Graphs. *ACM Transactions on Algorithms*, 21(4):1–14, 2025. Preliminary version in ICALP 2023.

30 Mohsen Ghaffari, Bernhard Haeupler, and Goran Zuzic. Hop-constrained oblivious routing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1208–1220, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451098.

31 Rohan Ghuge and Viswanath Nagarajan. Quasi-polynomial algorithms for submodular tree orienteering and directed network design problems. *Mathematics of Operations Research*, 47(2):1612–1630, 2022. doi:10.1287/MOOR.2021.1181.

32 Ashish Goel, Kajamalai G Ramakrishnan, Deepak Kataria, and Dimitris Logothetis. Efficient computation of delay-sensitive routes from one source to all destinations. In *Proceedings IEEE INFOCOM*, volume 2, pages 854–858. IEEE, 2001.

33 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. *SIAM Journal on Computing*, 52(2):298–322, 2022. Preliminary version in Proc. of STOC 2019.

34 Elena Grigorescu, Nithish Kumar, and Young-San Lin. Directed Buy-At-Bulk Spanners. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025)*, volume 353 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. `doi:10.4230/LIPIcs.APPROX/RANDOM.2025.22`.

35 Bernhard Haeupler, D. Ellis Hershkowitz, and Goran Zuzic. Tree embeddings for hop-constrained network design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 356–369, New York, NY, USA, 2021. Association for Computing Machinery. `doi:10.1145/3406325.3451053`.

36 Mohammad Taghi Hajiaghayi, Guy Kortsarz, and Mohammad R Salavatipour. Approximating buy-at-bulk and shallow-light k-steiner trees. *Algorithmica*, 53:89–103, 2009. `doi:10.1007/S00453-007-9013-X`.

37 Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 585–594, 2003. `doi:10.1145/780542.780628`.

38 Refael Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations research*, 17(1):36–42, 1992. `doi:10.1287/MOOR.17.1.36`.

39 D Ellis Hershkowitz and Richard Z Huang. Planar length-constrained minimum spanning trees. *arXiv preprint*, 2025. To appear in STOC 2026. `doi:10.48550/arXiv.2510.09002`.

40 D Ellis Hershkowitz and Richard Z Huang. Simple length-constrained minimum spanning trees. In *2025 Symposium on Simplicity in Algorithms (SOSA)*, pages 341–349, 2025. `doi:10.1137/1.9781611978315.25`.

41 Rhea Jain. *Approximation Algorithms for Connectivity and Fault-Tolerant Network Design*. University of Illinois at Urbana-Champaign, 2026.

42 David S Johnson and Michael R Garey. *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman, 1979.

43 Hans C Joksch. The shortest route problem with constraints. *Journal of Mathematical analysis and applications*, 14(2):191–197, 1966.

44 Erez Kantor and David Peleg. Approximate hierarchical facility location and applications to the shallow steiner tree and range assignment problems. In Tiziana Calamoneri, Irene Finocchi, and Giuseppe F. Italiano, editors, *Algorithms and Complexity*, pages 211–222, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. `doi:10.1007/11758471_22`.

45 Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Embeddings of planar quasimetrics into directed ℓ_1 and polylogarithmic approximation for directed sparsest-cut. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 480–491. IEEE, 2021. `doi:10.1109/FOCS52979.2021.00055`.

46 M. Reza Khani and Mohammad R. Salavatipour. Improved approximations for buy-at-bulk and shallow-light k-steiner trees and $(k, 2)$-subgraph. *J. Comb. Optim.*, 31(2):669–685, February 2016. `doi:10.1007/s10878-014-9774-5`.

47 Jochen Könemann, Asaf Levin, and Amitabh Sinha. Approximating the degree-bounded minimum diameter spanning tree problem. *Algorithmica*, 41:117–129, 2005. `doi:10.1007/S00453-004-1121-2`.

48 Guy Kortsarz and Zeev Nutov. Approximating some network design problems with node costs. *Theoretical Computer Science*, 412(35):4482–4492, 2011. `doi:10.1016/j.tcs.2011.04.013`.

49 Guy Kortsarz and David Peleg. Approximating shallow-light trees. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '97, pages 103–110, USA, 1997. Society for Industrial and Applied Mathematics.

50 Bundit Laekhanukit. On the integrality gap of directed Steiner tree LPs with relatively integral solutions. *arXiv preprint*, 2024. `arXiv:2412.10744`.

51 Shi Li and Bundit Laekhanukit. Polynomial integrality gap of flow LP for directed Steiner tree. *ACM Transactions on Algorithms*, 21(1):1–9, 2024. `doi:10.1145/3681791`.

52 Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979. doi:10.1137/0136016.

53 Dean H Lorenz and Danny Raz. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5):213–219, 2001. doi:10.1016/S0167-6377(01)00069-4.

54 Madhav V Marathe, R Ravi, Ravi Sundaram, S.S Ravi, Daniel J Rosenkrantz, and Harry B Hunt. Bicriteria network design problems. *Journal of Algorithms*, 28(1):142–171, 1998. doi:10.1006/jagm.1998.0930.

55 Hasan Pirkul and Samit Soni. New formulations and solution procedures for the hop constrained network design problem. *European Journal of Operational Research*, 148(1):126–140, 2003. doi:10.1016/S0377-2217(02)00366-1.

56 R. Ravi. Rapid rumor ramification: approximating the minimum broadcast time. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 202–213, 1994. doi:10.1109/SFCS.1994.365693.

57 Thomas Rothvoß. Directed Steiner tree and the Lasserre hierarchy. *CoRR*, abs/1111.5473, 2011. arXiv:1111.5473.

58 Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *J. ACM*, 51(6):993–1024, November 2004. doi:10.1145/1039488.1039493.

59 Arthur Warburton. Approximation of pareto optima in multiple-objective, shortest-path problems. *Operations research*, 35(1):70–79, 1987. doi:10.1287/OPRE.35.1.70.

60 Alexander Zelikovsky. A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica*, 18(1):99–110, 1997. doi:10.1007/BF02523690.

61 Leonid Zosin and Samir Khuller. On directed Steiner trees. In *Proceedings of ACM-SIAM SODA*, pages 59–63, 2002. URL: http://dl.acm.org/citation.cfm?id=545381.545388.