A Polylogarithmic Approximation for Directed Steiner Forest in Planar Digraphs

Chandra Chekuri * Rhea Jain †

Abstract

We consider Directed Steiner Forest (DSF), a fundamental problem in network design. The input to DSF is a directed edge-weighted graph $G = (V, E)$ and a collection of vertex pairs $\{(s_i, t_i)\}_{i\in[k]}$. The goal is to find a minimum cost subgraph H of G such that H contains an s_i-t_i path for each $i \in [k]$. DSF is NP-Hard and is known to be hard to approximate to a factor of $\Omega(2^{\log^{1-\epsilon}(n)})$ for any fixed $\epsilon > 0$ [17]. DSF admits approximation ratios of $O(k^{1/2+\epsilon})$ [10] and $O(n^{2/3+\epsilon})$ [4].

In this work we show that in planar digraphs, an important and useful class of graphs in both theory and practice, DSF is much more tractable. We obtain an $O(\log^6 k)$-approximation algorithm via the junction tree technique. Our main technical contribution is to prove the existence of a low density junction tree in planar digraphs. To find an approximate junction tree we rely on recent results on rooted directed network design problems [24, 11], in particular, on an LP-based algorithm for the Directed Steiner Tree problem [11]. Our work and several other recent ones on algorithms for planar digraphs [24, 36, 11] are built upon structural insights on planar graph reachability and shortest path separators [41].

*Dept. of Computer Science, Univ. of Illinois, Urbana-Champaign, Urbana, IL 61801. chekuri@illinois.edu. Supported in part by NSF grants CCF-1910149, CCF-1907937, and CCF-2402667.

†Dept. of Computer Science, Univ. of Illinois, Urbana-Champaign, Urbana, IL 61801. rheaj3@illinois.edu. Supported in part by NSF grants CCF-1907937 and CCF-2402667.

1 Introduction

Network design is a rich field of study in algorithms and discrete optimization. Problems in this area are motivated by various practical applications and have also been instrumental in the development of important tools and techniques. Two fundamental problems in network design are Steiner Tree and Steiner Forest. In Steiner Forest, the input is a graph $G = (V, E)$ with non-negative edge costs $c : E \to \mathbb{R}_+$ and a collection of vertex pairs $D = \{(s_i, t_i)\}_{i \in [k]}$; the goal is to find a subgraph $H \subseteq G$ of minimum cost such that for each $i \in [k]$, there exists an s_i-t_i path in H. The Steiner Tree problem is a special case of Steiner Forest in which there exists some root $r \in V$ such that $s_i = r$ for all $i \in [k]$; in other words, the goal is to connect a single source to a given set of sink vertices. When the input graph is *undirected*, these problems are both NP-Hard and APX-hard to approximate, and also admit constant-factor approximation algorithms; see Section 1.2 for a detailed discussion.

This paper considers the setting in which the input graph is *directed*. In Directed Steiner Forest (DSF), the input is a directed graph $G = (V, E)$ with edge-costs $c : E \to \mathbb{R}_+$, and the goal is to find a min-cost subgraph that contains a *directed path* (dipath) from s_i to t_i for each $i \in [k]$. Directed Steiner Tree (DST) is the special case when there is a single source r that needs to be connected to the sinks $t_i, i \in [k]$. In many settings directed graph problems tend to be more difficult to handle. Unlike their undirected counterparts, DST and DSF have strong lower bounds on their approximability. DST is known to be hard to approximate to a factor of $\Omega(\log^{2-\epsilon}(k))$ unless NP has randomized quasi-polynomial time algorithms [32], and to a factor $\Omega(\log^2 k/\log\log k)$ under other complexity assumptions [29] (see Section 1.2). Furthermore, a natural cut-based LP relaxation (used in approximation algorithms for several undirected network design problems) has a polynomial factor integrality gap; $\Omega(\sqrt{k})$ [44] or $\Omega(n^\delta)$ for some fixed $\delta > 0$ [38]. The best known approximation ratios for DST are $O(k^\epsilon)$ for any fixed $\epsilon > 0$ in polynomial time [43] and $O(\log^2 k/\log\log k)$ in *quasi-polynomial* time [29]. Whether DST admits a polynomial time poly-logarithmic approximation ratio has remained a challenging open problem for over 25 years.

The Directed Steiner Forest problem, on the other hand, is known *not* to admit a polylogarithmic approximation ratio unless $\mathsf{P} = \mathsf{NP}$. This is because DSF is hard to approximate to a factor $\Omega(2^{\log^{1-\epsilon}(n)})$ for any $\epsilon > 0$ via a simple reduction from the Label-Cover problem [17]. One technical reason for this difficulty is that, in spite of the name, a minimal feasible solution to DSF may not be a forest (unlike the case of undirected graphs). Note the contrast here to DST, in which a minimal feasible solution is an out-tree rooted at r. Due to this lack of structure as well as aforementioned hardness results, there has been limited progress in the development of approximation algorithms. The current best approximation ratios for DSF are $O(k^{1/2+\epsilon})$ [10] in the regime when k is small, and $O(n^{2/3+\epsilon})$ [4] when k is large; both results were obtained over a decade ago.

Recently, Friggstad and Mousavi [24] made exciting progress on DST by obtaining a simple $O(\log k)$-approximation in *planar* digraphs. Note that this establishes a separation between the hardness of DST in general digraphs and in planar digraphs. A follow-up work by Chekuri et al. [11] shows that the cut-based LP relaxation for DST has an integrality gap of $O(\log^2 k)$ in planar digraphs, which is in sharp contrast to the known lower bounds in general digraphs. Motivated by these positive results, as well as the inherent practical interest of planar graphs, we consider Directed Steiner Forest in planar digraphs (planar-DSF), and prove that it admits a poly-logarithmic approximation ratio.

Theorem 1.1. *There is an $O(\log^6 k)$-approximation for Directed Steiner Forest in planar digraphs, where k is the number of terminal pairs.*

Remark 1.1. (Node Weights) *Our algorithm and analysis generalize relatively easily to the setting in which both edges and nodes have non-negative weights. This is a consequence of the technique. The standard reduction of node-weighted problems to edge-weighted problems in directed graphs does not necessarily preserve planarity. Node-weighted Steiner pr oblems have also been considered separately in the undirected setting; see Section 1.2.*

1.1 Technical Overview and Outline

We prove Theorem 1.1 by employing the so-called *junction tree* technique. This technique allows one to reduce a multicommodity problem (such as DSF) to its rooted/single-source counterpart (such as DST). The power of this technique comes from the fact that in several network design problems, the single-source problem is often easier to solve. Junction-based schemes were initially highlighted in the context of non-uniform buy-at-bulk network design [31, 9], although the basic idea was already implicitly used for DSF in [8]. The technique has since been used to make progress in a variety of network design problems including improvements to DSF (see Section 1.2).

The high-level idea is as follows: we say $H \subseteq G$ is a *partial solution* if it is a feasible solution for some subset of terminal pairs. We look for a low-density partial solution, where *density* is defined as ratio of the cost of the partial solution to the number of terminal pairs it contains. Using a standard iterative approach for covering problems, one can reduce the original problem to the min-density partial solution problem while losing an additional $O(\log k)$ factor in the approximation ratio. In general, finding a min-density partial solution may still be hard; therefore, we restrict our attention to well-structured solutions. We aim to find partial solutions that contain some "junction" vertex $r \in V$ through which many pairs connect. Formally, in the context of DSF, we say a *junction tree* on terminal pairs $D_H \subseteq D$ is a subgraph $H \subseteq G$ with a root r such that for every terminal pair $(s_i, t_i) \in D_H$, H contains an s_i-r path *and* an r-t_i path.[1] The *density* of H is $c(H)/|D_H|$. The proof of Theorem 1.1 proceeds in two steps; first, we show that there exists a junction tree of low density, and second, we provide an algorithm to efficiently find a low-density junction tree given that one exists.

The key technical contribution of this paper is the first step: showing that any instance of planar-DSF contains a low-density junction tree. We prove the following Theorem in Section 2:

Theorem 1.2. *Given an instance (G, D) of planar-DSF, there exists a junction tree of density $O(\log^2 k)\mathrm{OPT}/k$ in G where $k = |D|$ and* OPT *is the cost of an optimum solution for (G, D).*

We prove the preceding theorem by considering an optimum solution $E^* \subseteq E$; we find several junction trees in E^* that are mostly disjoint and, in total, cover a large fraction of terminal pairs.

Remark 1.2. *We note that this proof strategy shows that there exists a low-density junction tree with respect to the optimal* integral *solution. It is an interesting open problem to prove a poly-logarithmic factor upper bound on the integrality gap of the natural cut-based LP relaxation for planar-DSF.*

We employ two tools developed by Thorup [41] on directed planar graphs in his work on reachability and approximate shortest path oracles. The first is a "layering" of a directed graph such that every path is contained in at most two consecutive layers, and each layer contains some nice tree-like structure. This allows us to restrict our attention to two layers at a time. We remark that a similar layering approach was recently used by [36] to obtain improved upper bounds on the multicommodity flow-cut gap in directed planar graphs; this was partly the inspiration for this work. The second tool is a "separator" theorem (essentially proved in [39], but given explicitly in [41]), which states that every planar graph contains three short paths whose removal results in connected components each containing at most half the number of vertices. We use this shortest-path separator to devise a recursive approach similar to that of [41]. We then restrict attention to one level of recursion in which many s_i-t_i paths pass through the separator, and show that in this case, we can use the nodes on the separator as roots for low-density junction trees.

For the second step, we prove the following theorem in Section 3:

Theorem 1.3. *Given an instance (G, D) of planar-DSF, there exists an efficient algorithm to obtain a junction tree of G of density at most $O(\log^3 k)$ times the optimal junction tree density in G.*

We show that any approximation algorithm for planar-DST with respect to the optimal *fractional* solution to an LP relaxation can be used to derive an approximation algorithm for finding the min-density junction tree. This uses a standard bucketing and scaling argument, initially given in the context of junction trees for buy-at-bulk network design [9]. This approach crucially relies on the fact that there exists a good approximation algorithm for planar-DST with respect to the natural LP relaxation. Finding junction trees without using the LP is not as straight forward. It is also not enough to be able to solve the min-density Directed Steiner Tree problem; there exist algorithms to do so in planar graphs via purely combinatorial techniques [11], however this approach does not extend to finding a good density junction tree due to the additional requirement that for each $(s_i, t_i) \in D_H$, we need to connect *both* s_i and t_i to r.

[1] This definition of junction tree does not necessarily correspond to a tree in a digraph; the terminology originated from the undirected setting.

1.2 Related Work **DST in general digraphs:** Directed Steiner Tree was first studied in approximation by Zelikovsky [43], who obtained an $O(k^\epsilon)$-approximation for any fixed $\epsilon > 0$. Charikar et al. [8] built on ideas from [43] to devise an $O(\log^3 k)$-approximation in quasi-polynomial time. This was later improved to $O(\log^2 k/\log\log k)$ in quasi-polynomial time, by Grandoni et al. [29] who used an LP-based approach, and by Ghuge and Nagarajan [26] who used a recursive greedy approach building on ideas in [12]. In terms of hardness, it is not difficult to see that DST generalizes Set Cover and is therefore hard to approximate to a factor $(1-\epsilon)\log k$ [19]; in fact, it is hard to approximate to a factor $\Omega(\log^{2-\epsilon}(k))$ unless NP has randomized quasipolynomial time algorithms [32]. Grandoni et al. [29] recently showed that even with quasi-polynomial time algorithms, DST is not approximable within a factor $\Omega(\log^2 k/\log\log k)$ unless the Projection Games Conjecture fails or $\mathsf{NP} \subseteq \mathsf{ZPTIME}(2^{n^\delta})$ for some $\delta \in (0,1)$. DST and algorithmic ideas for it are closely related to those for Group Steiner tree (GST) and Polymatroid Steiner tree (PST). We refer the reader to some relevant papers [25, 44, 10, 7, 11] for more details.

DSF and Junction Schemes: The first nontrivial approximation for Directed Steiner Forest was an $\tilde{O}(k^{2/3})$-approximation given by Charikar et al. [8]. 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 $O(k^{\frac{1}{2}+\epsilon})$ [10]; [10] 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 provide 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. [21] obtained an $O(n^\epsilon \cdot \min(n^{4/5}, m^{2/3}))$-approximation using a junction-based approach. This analysis was refined by [4] using ideas developed for finding good directed spanners, giving an improved approximation ratio of $O(n^{2/3+\epsilon})$. DSF with *uniform* edge costs admits an $O(n^{3/5+\epsilon})$-approximation [13].

DSF and DST have also been considered from a parameterized complexity perspective. DST is fixed parameter tractable parameterized by the number of terminals [18]. On the other hand, DSF is $W[1]$-hard [30]; however, it is polynomial time solvable if the number of terminals k is constant [20, 22].

Undirected Graphs: Steiner Tree admits a simple 2-approximation by taking a minimum spanning tree on the terminal set. There has been a long line of work improving this approximation factor using greedy techniques [42, 3, 35, 33], culminating in a $\left(1+\frac{\ln 3}{2}\right)$-approximation given by Robins and Zelikovsky [40]. This remained the best known approximation ratio for several years, until Byrka et al. developed an LP-based $(\ln 4+\epsilon)$-approximation [6, 27]. The Steiner Tree problem is APX-hard to approximate; in fact, there is no approximation factor better than $\frac{96}{95}$ unless $\mathsf{P} = \mathsf{NP}$ [14]. The Steiner Forest problem in undirected graphs admits a 2-approximation via primal-dual techniques [1, 28] and iterated rounding [34]. The node weighted versions of Steiner Tree and Steiner Forest admit an $O(\log k)$-approximation where k is the number of terminals [37], and further this ratio is asymptotically tight via a reduction from Set Cover.

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 [5]; this was later extended to a PTAS for Steiner Forest in graphs of bounded genus [2]. Recently, [15] 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 [16]. In directed graphs, along with the recent results discussed above, Friggstad and Mousavi obtained a constant-factor approximation for DST in minor-free graphs in the setting where the input graph is *quasi-bipartite* [23].

1.3 Definitions and Notation For a directed graph G, we let $V(G)$ and $E(G)$ denote the vertex and edge sets of G respectively. For $E' \subseteq E$, we let $V(E')$ denote the set of vertices in the graph induced by E'. For a subset $S \subseteq V$, we let $\delta^+(S)$ denote the set of all edges (u, v) with $u \in S$, $v \notin S$, and we let $\delta^-(S)$ denote the set of all edges (u, v) with $u \notin S$, $v \in S$. We will sometimes consider the *undirected* version of G; this is the underlying undirected graph obtained by ignoring orientations of edges in E.

For any directed path (dipath) $P \subseteq G$, and for any $u, w \in P$, we write $u <_P w$ if u appears before w in P. We define $>_P$, $\leq_P$, and $\geq_P$ similarly. For $u, w \in P$ with $u \leq_P w$, we let $P[u, w]$ denote the subpath of P from u to w. We denote the length of a path P, which is the number of edges in P, by $|P|$. For any $u, v, w \in G$, if P' is a u-v path and P'' is a v-w path, we let $P' \circ P''$ denote the concatenation of P'

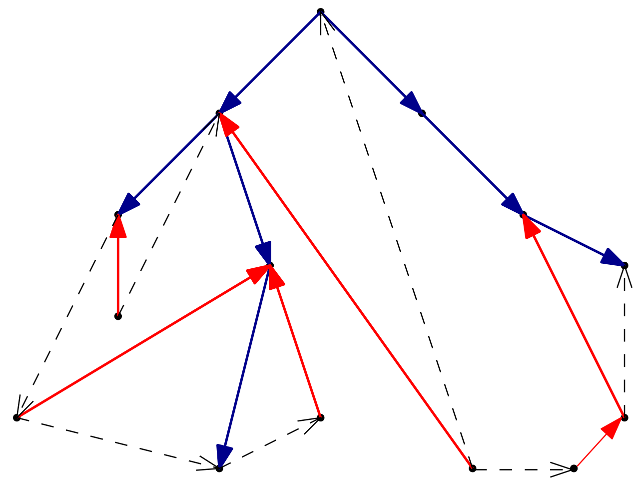

Figure 1: Example of a two-layered digraph. Bolded edges form the two-layered spanning tree; remaining edges in the graph are dashed. The two dipaths for each root to leaf path are denoted by blue and red edges: the first dipath away from the root given in blue and the second towards the root in red.

and P''. We will sometimes abuse notation and conflate path with dipath when it is clear from context. Unless explicitly stated, we do not distinguish between paths and walks, since we are only concerned with reachability.

Given an instance (G, D) of planar-DSF, we let OPT denote the value of an optimal solution.

Definition 1.1. (Junction Tree) *A* junction tree *on terminal pairs $D_H \subseteq D$ is a subgraph $H \subseteq G$ with a root r such that for every terminal pair $(s_i, t_i) \in D_H$, H contains an s_i-r path and an r-t_i path. The* density *of a junction tree is the ratio of its cost $c(H)$ to the number of terminal pairs $|D_H|$. We say a terminal pair $(s_i, t_i) \in D$ is* covered *by H if $(s_i, t_i) \in D_H$; that is, there exists an s_i-t_i walk in H containing r.*

For ease of notation, when considering subsets of terminal pairs $D' \subseteq D$, we sometimes write $i \in D'$ to mean $(s_i, t_i) \in D'$.

2 Existence of a good junction tree

This section proves Theorem 1.2, restated below:

Theorem 2.1. *Given an instance (G, D) of planar-DSF, there exists a junction tree of density $O(\log^2 k)\mathrm{OPT}/k$ in G where $k = |D|$ and OPT is the cost of an optimum solution for (G, D).*

Definition 2.1. *A* 2-layered spanning tree *of a digraph G is a rooted tree that is a spanning tree of the undirected version of G such that any path from the root to a leaf is the concatenation of at most 2 dipaths of G. A* 2-layered digraph *is a digraph that has a 2-layered spanning tree. The* root *of a 2-layered digraph is the root of its 2-layered spanning tree.*

Remark 2.1. *Note that a two-layered digraph may have additional edges aside from the spanning tree; we do not pose any restrictions on the directions of these edges. See Figure 1 for an example.*

The proof of Theorem 1.2 consists of three stages. First, in Section 2.1, we use a decomposition given by Thorup [41] of a directed graph into several 2-layered digraphs while preserving planarity. Using this decomposition, we show that it suffices to consider cases where the optimal solution is a 2-layered digraph; thus reducing proving Theorem 1.2 to proving Lemma 2.1:

Lemma 2.1. *Let (G, D) be an instance of planar-DSF. Suppose there exists a feasible solution $E^* \subseteq E(G)$ such that $G^* := (V(E^*), E^*)$ is a 2-layered digraph. Let r denote the root of G^*. Suppose that for each $(s_i, t_i) \in D$, there exists an s_i-t_i path in $G^* \setminus \{r\}$. Then there exists a junction tree $H \subseteq G^* \setminus \{r\}$ of density at most $O(\log^2 k)c(E^*)/k$.*

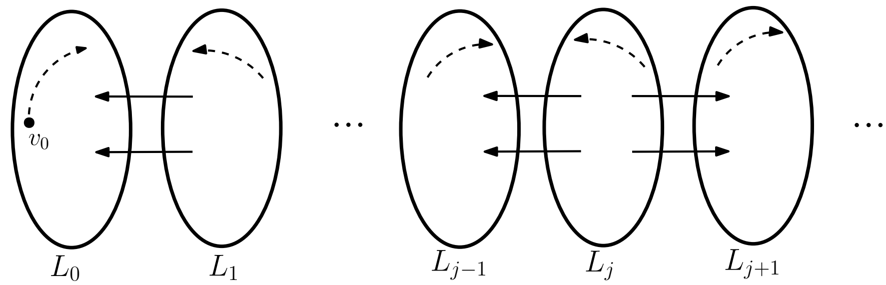

Figure 2: Layers constructed from G^*. Dotted lines represent edges inside each layer, while solid lines represent edges between layers. In this example, j is odd.

In Section 2.2, we use a recursive procedure built on a separator lemma on planar digraphs [41] to reduce proving Lemma 2.1 to Lemma 2.2:

LEMMA 2.2. *Let (G, D) be an instance of planar-DSF. Suppose there exists a feasible solution $E^* \subseteq E(G)$ that contains a dipath $P \subseteq E^*$ with the following property: every terminal pair $(s_i, t_i) \in D$ has a dipath $P_i \subseteq E^*$ from s_i to t_i such that $V(P) \cap V(P_i) \neq \emptyset$. Then E^* contains a junction tree of density at most $O(\log k)c(E^*)/k$.*

We conclude by proving Lemma 2.2 in Section 2.3.

2.1 Reduction to 2-Layered Digraphs In this section, we show that Lemma 2.1 suffices to prove Theorem 1.2, thus reducing to the case where the optimal solution is a 2-layered digraph. Let (G, D) be an instance of planar-DSF, and let $G^* = (V^*, E^*)$ be an optimal feasible solution of cost OPT. We assume without loss of generality that E^* induces a weakly connected graph; if not, we apply this decomposition on each weakly connected component separately. We use a decomposition of digraphs given by Thorup [41]. We include the details and proofs here for the sake of completeness, and to highlight some additional properties that we need. Let $v_0 \in V^*$ be an arbitrary node in G^*. We let L_0 be the set of all nodes in V^* that are reachable from v_0 in G^*. Then, we define alternating "layers" as follows:

$$L_j = \begin{cases} \{v \in V^* \setminus \cup_{j'<j} L_{j'} : v \text{ can reach } L_{j-1} \text{ in } G^*\} & j \text{ is odd} \\ \{v \in V^* \setminus \cup_{j'<j} L_{j'} : v \text{ is reachable from } L_{j-1} \text{ in } G^*\} & j \text{ is even} \end{cases}.$$

We continue this process until all vertices in V^* are covered by a layer; see Figure 2. Let ℓ denote the index of the last layer. For $j \in \{0, \ldots, \ell - 1\}$, we define G_j to be the graph obtained from G^* by deleting all nodes in $\cup_{i>j+1} L_i$ and contracting all nodes in $\cup_{i<j} L_j$. We call this contracted node the *root* r_j of G_j. It is clear from construction that each G_j is a 2-layered digraph. Furthermore, each G_j is a minor of G^* and is thus planar.[2]

CLAIM 2.1. *The total cost $\sum_{j=0}^{\ell-1} c(E(G_j)) \leq 2c(E^*)$.*

Proof. We show that each edge of E^* appears in at most two of the graphs from $G_0, \ldots, G_{\ell-1}$. Since $E(G_j) \subseteq E^*$ for all j, the claim follows. Let $(u, v) \in E^*$. If u, v are in the same layer L_j, then (u, v) is only in G_j and G_{j-1}; all other graphs $G_{j'}$ either contract (when $j' > j$) or delete (when $j' < j - 1$) L_j. If u, v are in distinct layers, they must be in adjacent layers L_j and L_{j+1}. For $j' > j + 1$, both L_j and L_{j+1} are contracted into the root, thus $(u, v) \notin G_{j'}$. For $j' < j$, L_{j+1} is deleted, thus once again $(u, v) \notin G_{j'}$. Therefore the edge (u, v) can only appear G_j and/or G_{j+1}. □

CLAIM 2.2. *For each pair $(s_i, t_i) \in D$, there exists some $j \in \{0, \ldots, \ell - 1\}$ such that $L_j \cup L_{j+1}$ contains an s_i-t_i path.*

[2]A graph H is a *minor* of G if it can be obtained from G by deleting and/or contracting edges of G. It is easy to see that if G is planar, then any minor of G is planar as well.

Proof. Let P_i be an s_i-t_i path in E^*; such a path must exist by feasibility of E^*. Let j be the minimum index such that L_j intersects P_i, and let v be a node in $L_j \cap P_i$.

If j is even, any node reachable from L_j must be contained in $\cup_{j' \leq j} L_{j'}$; thus $P_i[v, t_i] \subseteq \cup_{j' \leq j} L_{j'}$. By definition, L_{j+1} contains all nodes in $G^* \setminus \cup_{j' \leq j} L_{j'}$ that can reach L_j; thus $P_i[s_i, v]$ must be contained in $\cup_{j' \leq j+1} L_{j'}$.

Otherwise, if j is odd, any node that can reach L_j must be contained in $\cup_{j' \leq j} L_{j'}$, so $P_i[s_i, v] \subseteq \cup_{j' \leq j} L_{j'}$. L_{j+1} contains all nodes in $G^* \setminus \cup_{j' \leq j} L_{j'}$ reachable from L_j, so $P_i[v, t_i] \subseteq \cup_{j' \leq j+1} L_{j'}$.

In either case, $P_i \subseteq \cup_{j' \leq j+1} L_{j'}$. Since j is the minimum index that intersects P_i, $P_i \subseteq L_j \cup L_{j+1}$ as desired. ∎

Proof. [Reduction from Theorem 1.2 to Lemma 2.1] We partition the demand pairs D into $D_0, \ldots, D_{j-1}$, where $(s_i, t_i) \in D_j$ if $L_j \cup L_{j+1}$ contains an s_i-t_i path; if there are multiple such j we choose one arbitrarily. Note that all terminal pairs are covered by this partition by Claim 2.2.

Since $\sum_{j=0}^{\ell-1} c(E(G_j)) \leq 2c(E^*)$ and $D_0, \ldots, D_{j-1}$ form a complete partition of D, there must be some $j \in \{0, \ldots, \ell - 1\}$ such that $c(E(G_j))/|D_j| \leq 2c(E^*)/|D|$. We claim that (G_j, D_j) satisfies the conditions of Lemma 2.1: G_j is a planar 2-layered digraph that is a feasible solution on all terminal pairs D_j, and for each $i \in D_j$, there is an s_i-t_i path contained in $L_j \cup L_{j+1}$, thus avoiding the root of G_j. By Lemma 2.1, there exists a junction tree H in G_j of density

$$O(\log^2 |D_j|) \frac{c(E(G_j))}{|D_j|} \leq O(\log^2 k) \frac{2c(E^*)}{|D|} = O(\log^2 k)\mathrm{OPT}/k.$$

Furthermore, since H does not contain the root of G_j, H is a subgraph of G^*. ∎

2.2 Reduction from Two-Layered Digraphs to One-Path Setting In this section, we show that assuming Lemma 2.2, we can prove Lemma 2.1, restated below:

LEMMA 2.1. *Let (G, D) be an instance of planar-DSF. Suppose there exists a feasible solution $E^* \subseteq E(G)$ such that $G^* := (V(E^*), E^*)$ is a 2-layered digraph. Let r denote the root of G^*. Suppose that for each $(s_i, t_i) \in D$, there exists an s_i-t_i path in $G^* \setminus \{r\}$. Then there exists a junction tree $H \subseteq G^* \setminus \{r\}$ of density at most $O(\log^2 k)c(E^*)/k$.*

Fix an instance (G, D) of planar-DSF and a feasible solution E^* satisfying the conditions outlined in the statement of Lemma 2.1 above. Let $T^* \subseteq E^*$ be a 2-layered spanning tree with root r. Given any *undirected* tree T, we let $P_T(u, v)$ denote the unique tree path from u to v. We will follow a recursive process to partition D into subsets on which we build junction trees. To do so, we use the following separator lemma on planar digraphs.

LEMMA 2.3. ([41]) *Given an* undirected *planar graph $G = (V, E)$ with a spanning tree T rooted at r and non-negative vertex weights $w : V \to \mathbb{R}_{\geq 0}$, we can find three vertices u_1, u_2, u_3 such that each component of $G \setminus (P_T(r, u_1) \cup P_T(r, u_2) \cup P_T(r, u_3))$ has at most half the weight of G.*

We define vertex weights $w(v) = 1$ if $v \in D$ and $w(v) = 0$ otherwise. We consider the undirected version of E^*; that is, we ignore all directions on E^* and apply Lemma 2.3 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 a 2-layered spanning tree, each path $P_{T^*}(r, u_i)$ consists of at most 2 dipaths of E^*. We remove the root r and let Q_i^1, Q_i^2 denote the at most two dipaths of $P_{T^*}(r, u_i) \setminus \{r\}$. Let $S_0 = \cup_{i \in [3]} \{Q_i^1, Q_i^2\}$ denote this set of at most 6 dipaths; we call this a *separator*. We define $D_0 \subseteq D$ to be the set of all terminal pairs (s_i, t_i) such that E^* contains an s_i-t_i path going through one of the dipaths in S_0. Equivalently, $(s_i, t_i) \in D_0$ iff there exists an s_i-t_i path $P_i \subseteq E^*$ such that $V(P_i) \cap V(S_0) \neq \emptyset$. See Figure 3 for an example of a separator with the corresponding set D_0. We let $\mathcal{C}_0$ be the set of weakly connected components of $G \setminus (\cup_{i \in [3]} P_{T^*}(r, u_i))$; we drop "weakly connected" and simply refer to these as "components" in the remainder of this section. Note that each $C \in \mathcal{C}_0$ has at most half the total number of terminals.

We recurse on each component $C \in \mathcal{C}_0$ as follows: we contract S_0 into r and recurse on the sub-instance consisting of C and the new contracted root r. It is not difficult to see that this new sub-instance is a 2-layered digraph and thus contains a 2-layered spanning tree T_C^*. We repeat the same process as above,

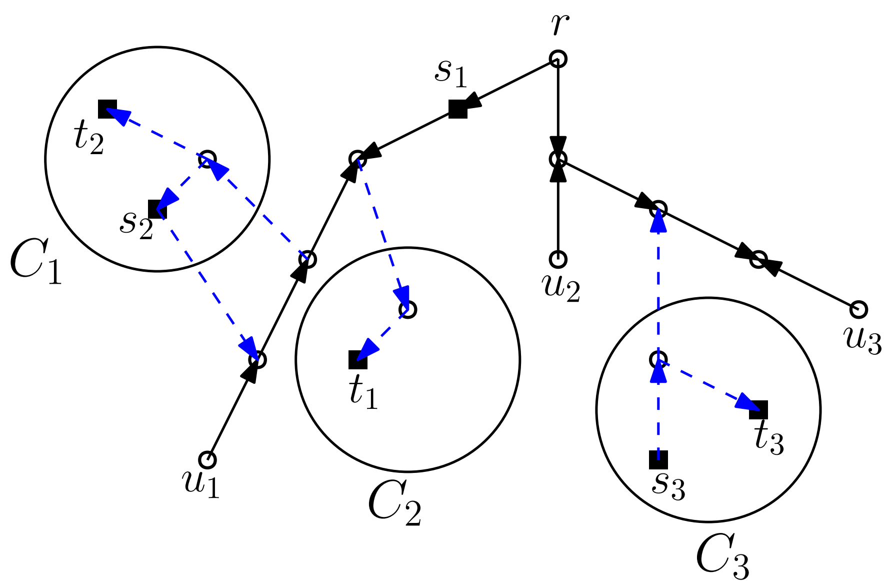

Figure 3: Example of separator and resulting weakly connected components. Solid black lines denote edges in the separator S_0, while dashed blue lines represent edges between components and the separator. Terminals are labeled and denoted with boxes. In this example, $D_0 = \{(s_1, t_1), (s_2, t_2)\}$ since there exists an s_1-t_1 and an s_2-t_2 path through the separator. Notice that $(s_2, t_2) \in D_0$ even though s_2 and t_2 remain in the same component C_1.

applying Lemma 2.3 with T_C^*, and weights the same as before for all nodes in C and $w(r) = 0$. We obtain three nodes $u_1', u_2', u_3' \in C$. Once again, we ignore r when considering the dipaths. We define S_1^C to be the set of at most 6 dipaths in $\cup_{i\in[3]}(P_T(r, u_i') \setminus \{r\})$, and let D_1^C be the set of all $(s_i, t_i) \in D$ with $s_i, t_i \in C$ such that there exists an s_i-t_i path in C with a non-empty intersection with a dipath in S_1^C.

Remark 2.2. *In the recursive step, we choose to contract the separator into the root to maintain the property that each recursive call still corresponds to a 2-layered digraph. It is important to remove the contracted root r from the dipaths of S_i, $i > 0$ to ensure that all nodes in the separator are nodes in G and all separators are disjoint. We remove the root r from the dipaths of S_0 to ensure that H does not contain the root of E^*, in order to satisfy the lemma statement.*

We continue this recursive process until each component has at most one terminal. Since the number of terminals halve at each step, the total recursion depth is at most $\lceil \log 2k \rceil = \lceil \log k \rceil + 1$. For ease of notation, for $j \geq 1$ we denote by $S_j := \cup_{C\in\mathcal{C}_{j-1}} S_j^C$ the set of all dipaths constructed in the jth level of recursion and let $D_j := \cup_{C\in\mathcal{C}_{j-1}} D_j^C$.

Claim 2.3. $D \subseteq \cup_{j\in 0}^{\lceil \log k\rceil+1} D_j$.

Proof. Fix $(s_i, t_i) \in D$, and let P_i be an s_i-t_i path in $E^* \setminus \{r\}$. Let j be the first recursive level such that P_i intersects S_j; such a level must exist since by the last step of recursion, s_i and t_i are in different components. Let C be the component such that P_i intersects S_j^C. Then P_i must be fully contained in C; else P_i would have intersected a separator at an earlier level. Thus $(s_i, t_i) \in D_j^C \in D_j$. ∎

Corollary 2.1. *There exists a recursion level $j^* \in \{0, \ldots, \lceil \log k\rceil + 1\}$ such that $|D_{j^*}| \geq \frac{k}{\lceil \log k\rceil+2}$.*

Corollary 2.1 allows us to focus on one recursion layer that covers a large number of terminal pairs, and use the planar separators S_{j^*} to reduce to the one path case.

Proof. [Reduction from Lemma 2.1 to Lemma 2.2] Let j^* be the recursion level given by Corollary 2.1 such that $|D_{j^*}| \geq \frac{k}{\lceil \log k\rceil+2}$. Recall that we define $\mathcal{C}_{j^*}$ to be the set of all components at level j^*. Note that

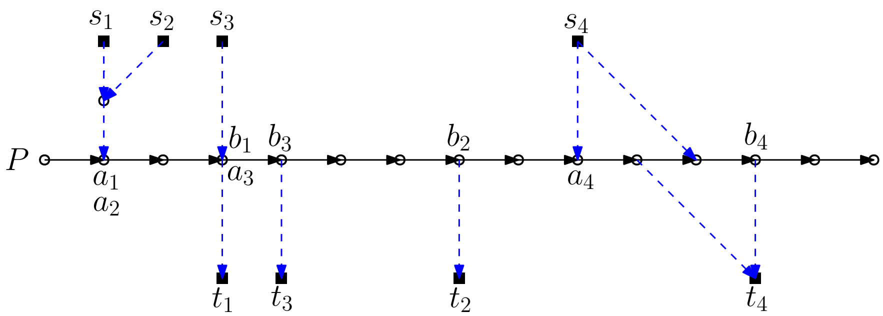

Figure 4: The path P is given with solid black lines. Blue dashed lines represent the paths between terminals and P. Note that terminals can have multiple paths to/from P, as shown by s_4/t_4. In this example, terminal pairs $1, 2, 3$ all have mutually overlapping intervals and thus form a junction tree rooted at the vertex $b_1 = a_3$.

all components $C \in \mathcal{C}_j^*$ are disjoint; therefore, $\sum_{C\in\mathcal{C}_{j^*}} c(E(C)) \leq c(E^*)$. Furthermore, since $(s_i, t_i) \in D_{j^*}^C$ implies that $s_i, t_i \in C$, $D_{j^*}^C$ form a partition of D_{j^*}. Thus there must be one component $C \in \mathcal{C}_{j^*}$ such that $c(E(C))/|D_{j^*}^C| \leq c(E^*)/|D_{j^*}|$. Fix this component C.

By construction, for all $(s_i, t_i) \in D_{j^*}^C$ there is an s_i-t_i path intersecting $S_{j^*}^C$ that is fully contained in $C \setminus \{r\}$. Since $S_{j^*}^C$ consists of at most 6 dipaths, there must be at least one dipath, which we call $Q_{j^*}^C$, such that at least $\frac{1}{6}$ of the terminal pairs in $D_{j^*}^C$ have paths that intersect $Q_{j^*}^C$; we denote this subset of terminal pairs by D^*. We apply Lemma 2.2 on $(C \setminus \{r\}, D^*)$ to obtain a junction tree H of density at most

$$O(\log |D^*|)\frac{c(C)}{|D^*|} \leq O(\log k)\frac{6c(C)}{|D_{j^*}^C|} \leq O(\log k)\frac{6c(E^*)}{|D_{j^*}|} \leq O(\log^2 k)c(E^*)/k.$$

□

2.3 One-Path Setting In this section we prove Lemma 2.2, restated below:

Lemma 2.2. *Let (G, D) be an instance of planar-DSF. Suppose there exists a feasible solution $E^* \subseteq E(G)$ that contains a dipath $P \subseteq E^*$ with the following property: every terminal pair $(s_i, t_i) \in D$ has a dipath $P_i \subseteq E^*$ from s_i to t_i such that $V(P) \cap V(P_i) \neq \emptyset$. Then E^* contains a junction tree of density at most $O(\log k)c(E^*)/k$.*

Fix an instance (G, D) of planar-DSF and a solution E^* with dipath $P \subseteq E^*$ satisfying the conditions outlined in the statement of Lemma 2.2 above. We will sometimes overload notation and write P as $V(P)$. We label the vertices on P as $v_0, \ldots, v_{|P|}$. For each terminal pair s_i-t_i, let a_i denote the first node in P that s_i can reach, and let b_i denote the last node in P that can reach t_i (here, reachability is defined using edges in E^*). By the condition in Lemma 2.2, $a_i \leq_P b_i$ for all $i \in [k]$; else no s_i-t_i path could intersect P. We let I_i denote the interval $P[a_i, b_i]$. We let P_{s_i} denote the path in E^* from s_i to a_i and let P_{t_i} denote the path in E^* from b_i to t_i. See Figure 4 for an example.

We start with a simple observation regarding these intervals and their relation to junction trees; we show that if there exists a set of intervals which all overlap at a common vertex, then we can form a junction tree on the corresponding terminal pairs.

Claim 2.4. *Let $D' \subseteq D$ such that $\cap_{i\in D'} I_i \neq \emptyset$, i.e. all intervals overlap. Let $a_{start} = \min_{i\in D'} a_i$ and $b_{end} = \max_{i\in D'} b_i$, where min and max are taken with respect to $\leq_P$. Then $H = P[a_{start}, b_{end}] \cup \bigcup_{i\in D'}(P_{s_i} \cup P_{t_i})$ is a valid junction tree on D'.*

Proof. Let v be some element in $\cap_{i\in D'} I_i$; this will be the root of the junction tree H. It suffices to show that for all $(s_i, t_i) \in D'$, s_i can reach v and v can reach t_i in H. Let $(s_i, t_i) \in D'$. By definition of a_{start} and b_{end}, and since $v \in I_i$, we have that $a_{\text{start}} \leq_P a_i \leq_P v \leq_P b_i \leq_P b_{\text{end}}$. Therefore $P[a_i, v]$ and $P[v, b_i]$ are contained in H. Thus the s_i-v path $P_{s_i} \circ P[a_i, v]$ is contained in H, as is the v-t_i path $P[v, b_i] \circ P_{t_i}$, as desired. □

Claim 2.4 provides a natural way to obtain junction trees in E^*: we partition D into groups such that in each group, all corresponding intervals overlap at a common vertex, and then form the junction trees accordingly. To partition D, we first separate terminal pairs based on their interval lengths; recall that path lengths are defined in terms of number of edges, so the length of the interval I_i is the number of edges from a_i to b_i in P. We let D_0 denote the set of all $(s_i, t_i) \in D$ such that $a_i = b_i$; these correspond to 0-length intervals. For $j \in \{1, \ldots, \log|P| + 1\}$, let $D_j = \{(s_i, t_i) : |I_i| \in [2^{j-1}, 2^j)\}$. For $v \in P$, we let $D_j^v \subseteq D_j$ be the set of all $(s_i, t_i) \in D_j$ such that $v \in I_i$. We construct the set of groups

$$\mathcal{G} = \{D_0^v : \exists i \in D_0 \text{ s.t. } a_i = b_i = v\} \cup \bigcup_{j \in [\log|P|+1]} \{D_j^{v_\ell} : \ell \text{ is a multiple of } 2^{j-1}\}.$$

Note that for each group D_j^v, $v \in \cap_{i \in D_j^v} I_i$. Therefore, each group D_j^v is associated with a junction tree H_j^v with root v as given by Claim 2.4. We let $\mathcal{H}$ denote the set of all such junction trees.

Claim 2.5. *Every $(s_i, t_i) \in D$ is in some group in $\mathcal{G}$.*

Proof. Fix $(s_i, t_i) \in D$. If $a_i = b_i$, then $(s_i, t_i) \in D_0$, so $(s_i, t_i) \in D_0^{a_i} \in \mathcal{G}$. Else, $|I_i| \in \{1, \ldots, |P|\}$, so $\exists j \in \{1, \ldots, \log|P| + 1\}$ such that $(s_i, t_i) \in D_j$. Let ℓ be the first multiple of 2^{j-1} such that $v_\ell \geq_P a_i$. Then $P[a_i, v_\ell] \leq 2^{j-1}$. Since $(s_i, t_i) \in D_j$, $|I_i| \geq 2^{j-1}$; thus $v_\ell \leq_P b_i$. Therefore $v_\ell \in I_i$, so $(s_i, t_i) \in D_j^{v_\ell} \in \mathcal{G}$. ∎

We will show that the junction trees in $\mathcal{H}$ have, on average, low density. To do so, one must ensure that each edge $e \in E^*$ only appears in $O(\log k)$ junction trees to maintain the cost bound. A technical difficulty is reasoning about the edges of $E^* \setminus P$, since the paths between terminals and the path P may intersect and share edges. The following key observation provides some structure on these paths with respect to the intervals:

Claim 2.6. *For any $i, i' \in [k]$, if $P_{s_i} \cap P_{s_{i'}} \neq \emptyset$, then $a_i = a_{i'}$. Similarly, $P_{t_i} \cap P_{t_{i'}} \neq \emptyset$, then $b_i = b_{i'}$.*

Proof. Consider $i, i' \in [k]$, and suppose without loss of generality that $a_i \leq_P a_{i'}$. Let $v \in P_{s_i} \cap P_{s_{i'}}$. Then $P_{s_{i'}}[s_{i'}, v] \circ P_{s_i}[v, a_i]$ is a path from $s_{i'}$ to a_i in E^*. Since we defined $a_{i'}$ as the earliest point that $s_{i'}$ can reach on P, it must be the case that $a_{i'} \leq_P a_i$, so $a_i = a_{i'}$. An analogous argument shows that for any $i, i' \in [k]$, if $P_{t_i} \cap P_{t_{i'}} \neq \emptyset$, then $b_i = b_{i'}$. ∎

Claim 2.7. *Each node in P appears in at most $5 \log|P| + 6$ junction trees in $\mathcal{H}$. The same holds for each edge in P.*

Proof. Let $u \in P$. By Claim 2.4, for any $H_j^v \in \mathcal{H}$, $H_j^v \cap P = P[a_{\text{start}}, b_{\text{end}}]$, where a_{start} is the first interval start point and b_{end} is the last interval end point of all intervals of D_j^v. Notice that since the intervals of D_j^v overlap at a common vertex, $P[a_{\text{start}}, b_{\text{end}}]$ is equivalent to $\cup_{i \in D_j{}^v} I_i$.

First, consider $j = 0$. In this case, for any v, $\cup_{i \in D_0^v} I_i = \{v\}$. Thus $u \in D_0^v$ if and only if $u = v$, so u is in at most one group when $j = 0$.

Next, fix $j \geq 1$, and consider some v_ℓ such that $D_j^{v_\ell} \in G$. Note that $(s_i, t_i) \in D_j^{v_\ell}$ implies that $|I_i| < 2^j$ and $v_\ell \in I_i$. Therefore, it must be the case that $a_i >_P v_{\ell - 2^j}$ and $b_i <_P v_{\ell + 2^j}$. Thus for any $D_j^{v_\ell} \in \mathcal{G}$, $\cup_{i \in D_j^{v_\ell}} I_i \subseteq P[v_{\ell - 2^j}, v_{\ell + 2^j}]$. Therefore if $u \in D_j^{v_\ell}$, then v_ℓ has to be within 2^j edges of u. Since $\mathcal{G}$ only contains $D_j^{v_\ell}$ for ℓ a multiple of 2^{j-1}, there are at most 5 values of ℓ that are multiples of 2^{j-1} such that v_ℓ can either reach or be reached by u within 2^j edges. Therefore, u is in at most 5 groups for any fixed j. Summing over all $j = 1, \ldots, \log|P| + 1$ gives the desired bound.

Each edge $e \in P$ is only in a junction tree H if both its endpoints are also in H. Thus the same upper bound holds for each edge in P. ∎

Claim 2.8. *Each node in $V(E^*) \setminus P$ appears in at most $10 \log|P| + 12$ junction trees in $\mathcal{H}$. The same holds for each edge in $E^* \setminus P$.*

Proof. Fix $u \in V(E^*) \setminus P$. If u is in any junction tree H, it must be in some P_{s_i} and/or some P_{t_i}; it may be in many such paths for various terminal pairs. Let $D_s = \{i : u \in P_{s_i}\}$ and $D_t = \{i : u \in P_{t_i}\}$.

By Claim 2.6, there exists some node $a \in P$ such that for all $i \in D_s$, $a_i = a$. Similarly, there exists some $b \in P$ such that for all $i \in D_t$, $b_i = b$. By construction of junction trees in Claim 2.4, for any $H \in \mathcal{H}$, $u \in H$ only if $a \in H$ or $b \in H$. By Claim 2.7, a and b are each in at most $5 \log |P| + 6$ junction trees in $\mathcal{H}$. Therefore, u is in at most $2(5 \log |P| + 6)$ junction trees in $\mathcal{H}$.

Each edge $e \in E^*$ is only in a junction tree H if both its endpoints are also in H. Thus the same upper bound holds for each edge in E^*. ∎

We conclude the proof of the main lemma:

Proof. [Proof of Lemma 2.2] By Claims 2.7 and 2.8, each edge of E^* is in at most $O(\log |P|)$ junction trees, so $\sum_{H \in \mathcal{H}} c(H) \leq O(\log |P|) c(E^*)$. We note that while $|P|$ could be as large as $\Theta(n)$, we can effectively assume $|P| \leq 2k$ as follows. First, we assume $E^* = P \cup_{i \in [k]} (P_{s_i} \cup P_{t_i})$; these are the only edges used in junction trees $\mathcal{H}$ and constitutes a feasible solution. Then, we can ignore all degree-2 nodes in P: if $v_i \in P$ has degree 2 in E^*, we can replace the edges $e' = (v_{i-1}, v_i)$ and $e'' = (v_i, v_{i+1})$ with an edge $e = (v_{i-1}, v_{i+1})$ of cost $c(e') + c(e'')$ without changing feasibility of E^*. The only nodes in P that have degree greater than 2 in E^* are the points a_i, b_i for $i \in [k]$. Thus we can assume $|P| \leq 2k$, and $\sum_{H \in \mathcal{H}} c(H) \leq O(\log k) c(E^*)$.

By Claim 2.5, all terminal pairs are covered by at least one junction tree in $\mathcal{H}$. Therefore, the total density of junction trees in $\mathcal{H}$ is $O(\log k) c(E^*)/k$. An averaging argument shows that there must be at least one $H^* \in \mathcal{H}$ that has density at most $O(\log k) c(E^*)/k$. ∎

3 Finding a good junction tree

In this section we show that there exists an efficient algorithm to find an approximate min-density junction tree, proving Theorem 1.3 restated below:

Theorem 3.1. *Given an instance (G, D) of planar-DSF, there exists an efficient algorithm to obtain a junction tree of G of density at most $O(\log^3 k)$ times the optimal junction tree density in G.*

We employ an LP-based approach. We consider a natural cut-based LP relaxation for DST with variables $x_e \in [0, 1]$ for $e \in E$ indicating whether or not e is in the solution. Here, the input is a digraph $G = (V, E)$ with root r and terminals t_i, $i \in [k]$.

$$\text{(DST-LP)} \qquad \begin{aligned} \min \quad & \sum_{e \in E} c(e) x_e \\ s.t. \quad & \sum_{e \in \delta^+(S)} x_e \geq 1 \quad \forall S \subseteq V, r \in S, \exists i \text{ s.t. } t_i \notin S \\ & x_e \geq 0 \quad \forall\, e \in E \end{aligned}$$

We prove the following lemma:

Lemma 3.1. *Suppose there exists an α-approximation for DST in planar graphs with respect to the optimal solution to DST-LP. Then, given a planar-DSF instance (G, D), there exists an efficient algorithm to obtain a junction tree of G of density at most $O(\alpha \cdot \log k)$ times the optimal junction tree density in G.*

It was recently shown by [11] that there exists an $O(\log^2 k)$ approximation for DST in planar graphs with respect to the optimal solution to DST-LP. Therefore it suffices to prove Lemma 3.1 to prove Theorem 1.3.

Let (G, D) be an instance to planar-DSF. We start by guessing the root r of the junction structure, as we can repeat this algorithm for each $r \in V$ and choose the resulting junction structure of minimum density. We consider the following LP relaxation for finding the minimum density junction tree rooted at r. We follow a similar structure to that of DST-LP, with additional variables y_{s_i} and y_{t_i} for each $i \in [k]$ to indicate whether or not s_i and t_i are included in the solution. We ensure that $y_{s_i} = y_{t_i}$ so that the junction tree includes complete pairs rather than individual terminals. We also change the direction of flow from each s_i to the root r. The resulting minimum density would be $(\sum_{e \in E} c(e) x_e)/(\sum_{i \in [k]} y_{t_i})$; we normalize

$\sum_{i\in[k]} y_{t_i} = 1$.

$$\text{(Den-LP)} \qquad \begin{aligned} \min \quad & \sum_{e\in E} c(e)x_e \\ s.t. \quad & \sum_{e\in\delta^+(S)} x_e \geq y_{t_i} \quad \forall i\in[k], \forall S\subseteq V, r\in S, t_i\notin S \\ & \sum_{e\in\delta^-(S)} x_e \geq y_{s_i} \quad \forall i\in[k], \forall S\subseteq V, r\in S, s_i\notin S \\ & y_{s_i} = y_{t_i} \quad \forall i\in[k] \\ & \sum_{i\in[k]} y_{t_i} = 1 \\ & x_e, y_{s_i}, y_{t_i} \geq 0 \quad \forall\, e\in E, i\in[k] \end{aligned}$$

We claim that Den-LP provides a valid lower bound for the optimum density of a junction tree through r.

Claim 3.1. *For any junction tree H of G, there exists a feasible fractional solution (x, y) to Den-LP such that $\sum_{e\in E} c(e)x_e = c(H)/|D_H|$, where D_H is the set of terminal pairs covered by H.*

Proof. Let H be any junction tree of G, let $D_H \subseteq D$ be the terminal pairs covered by H. Consider (x, y) given by $x_e = 1/|D_H|$ if $e \in H$ and 0 otherwise, $y_{t_i} = 1/|D_H|$ if $(s_i, t_i) \in D_H$ and 0 otherwise, and $y_{s_i} = y_{t_i}$ for all $i \in [k]$. For each $(s_i, t_i) \in D_H$, since H contains an s_i-r path and an r-t_i path, x supports a flow of $1/|D_H|$ from s_i to r and r to t_i; thus the first two sets of constraints are satisfied. It is easy to verify that the rest of the constraints are satisfied and that $\sum_{e\in E} c(e)x_e = c(H)/|D_H|$. ∎

Despite the fact that the LP has exponentially many constraints, it can be solved efficiently via a separation oracle: suppose we are given a fractional solution (x, y). The first two sets of constraints are satisfied if for every $i \in [k]$, x supports a flow of at least y_{t_i} from r to t_i and a flow of at least y_{s_i} from s_i to r; these can be checked via min-cut computations. There are only polynomially many remaining constraints; thus these can be checked in polynomial time. One can also write a compact LP via additional flow variables.

To find a junction tree of G, We first solve Den-LP to obtain an optimal fractional solution (x^*, y^*). For $j = 0, \ldots, \log k$, we let $D_j = \{(s_i, t_i) \in D : y_{t_i} \in (\frac{1}{2^{j+1}}, \frac{1}{2^j}]\}$. We will show that there exists a group $\theta \in \{0, \ldots, \log k\}$ for which the total y^* value is large; thus x^* supports a good fraction of flow from the root to/from D_θ.

Claim 3.2. *There exists $\theta \in \{0, \ldots, \log k\}$ such that $\sum_{i\in D_\theta} y_{t_i} \geq 1/(2\log k + 2)$.*

Proof. If $(s_i, t_i) \in D \setminus (\cup_{j=0}^{\log k} D_j)$, then $y_{t_i} \leq \frac{1}{2^{\log k+1}} = \frac{1}{2k}$. Therefore, the total y value of pairs not covered by the sets D_j is at most $\sum_{i\notin\cup_{j=0}^{\log k} D_j} y_{t_i} \leq k\frac{1}{2k} = \frac{1}{2}$. Since $\sum_{i\in[k]} y_{t_i} = 1$, the total y value of pairs covered by the sets D_j is at least $\sum_{i\in\cup_{j=0}^{\log k} D_j} y_{t_i} \geq \frac{1}{2}$. Since there are $\log k + 1$ disjoint groups, there is a group whose total y value is at least $1/(2(\log k + 1))$. ∎

Let θ be given by Claim 3.2. We use the α-approximation algorithm for DST twice: first, we consider the instance on G with terminal set $D^t_\theta = \{t_i : (s_i, t_i) \in D_\theta\}$ and obtain a directed r-tree T_t. Second, we let G' be obtained from G by reversing the direction of all edges. We apply the α-approximation algorithm for DST on G' with terminal set $D^s_\theta = \{s_i : (s_i, t_i) \in D_\theta\}$ and obtain a directed r-tree T_s in G'. Note that T_s is a directed in-tree in G; therefore, $T = T_t \cup T_s$ is a valid junction tree on G and terminal pairs D_θ.

Claim 3.3. *$2^{\theta+1}x^*$ is a feasible solution to DST-LP on both of the following instances:*

- *G with terminal set D^t_θ,*

- G' *with terminal set* D_θ^s.

Proof. We first consider G with terminal set D_θ^t. Fix $S \subseteq V$, $t_i \in D_\theta^t$ such that $r \in S, t_i \notin S$. Since x^* is a feasible solution to Den-LP and $i \in D_\theta$,

$$\sum_{e \in \delta_G^+(S)} 2^{\theta+1} x_e^* \geq 2^{\theta+1} y_{t_i} > 2^{\theta+1} \frac{1}{2^{\theta+1}} = 1.$$

Next, we consider G' with terminal set D_θ^s, and fix $S \subseteq V$ and $s_i \in D_\theta^s$ such that $r \in S, s_i \notin S$. Then, using the fact that $y_{s_i} = y_{t_i}$ for all i, we use the same argument as above:

$$\sum_{e \in \delta_{G'}^+(S)} 2^{\theta+1} x_e^* = \sum_{e \in \delta_G^-(S)} 2^{\theta+1} x_e^* \geq 2^{\theta+1} y_{s_i} = 2^{\theta+1} y_{t_i} > 2^{\theta+1} \frac{1}{2^{\theta+1}} = 1.$$

In both cases, all corresponding constraints in DST-LP are satisfied. ∎

Claim 3.4. *The density of* T *is at most* $O(\alpha \log k) \sum_{e \in E} c(e) x_e^*$.

Proof. By Claim 3.3, along with the fact that the DST algorithm used to construct T_t and T_s is an α-approximation with respect to the optimal fractional solution, the costs of T_t and T_s are each upper bounded by $\alpha 2^{\theta+1} \sum_{e \in E} c(e) x_e^*$.

To bound the number of terminals covered by T, i.e. $|D_\theta|$, note that $\sum_{i \in D_\theta} y_{t_i} \leq \sum_{i \in D_\theta} 1/2^\theta = |D_\theta| 1/2^\theta$. Thus by Claim 3.2, $|D_\theta| \geq 2^\theta \sum_{i \in D_\theta} y_{t_i} \geq 2^\theta/(2 \log k + 2)$. Therefore, the density of T is at most

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

∎

Lemma 3.1 follows from Claim 3.4, Claim 3.1, and the fact that T is a valid junction tree.

4 Proof of Theorem 1.1

Theorem 1.2 and Theorem 1.3 suffice to conclude the proof of Theorem 1.1 via a greedy covering approach that is standard for covering problems such as Set Cover.

Proof. [Proof of Theorem 1.1] Let (G, D) be an instance of planar-DSF. Combining Theorems 1.2 and 1.3, we can find a junction tree of density at most $O(\log^5 k)\mathrm{OPT}(G)/k$. Let H_1 be such a junction tree on (G, D), and let D_1 be the set of terminal pairs covered by H_1. We remove D_1 from D and repeat until all terminal pairs are covered. Since each junction tree covers at least one terminal pair, this process terminates in at most k iterations. Let $H_1, \ldots, H_\ell$ be the junction trees formed by this process, and for $j \in [\ell]$, let D_j be the set of terminals covered by H_j. We denote by $D_{<j}$ the set $\cup_{j'<j} D_{j'}$. We then return $H = \cup_{j \in [\ell]} H_j$.

It is clear by construction that H is a feasible solution: for each $i \in [k]$, there exists some junction tree H_j that covers (s_i, t_i), and the path in H_j from s_i to its root concatenated with the path from the root to t_i is an s_i-t_i path in H. To bound the cost of H, note that $c(H) \leq \sum_{j \in [\ell]} c(H_j) = \sum_{j \in [\ell]} |D_j| \cdot \mathrm{density}(H_j)$. By construction, each H_j has density at most $O(\log^5 k_j)\mathrm{OPT}(G)/k_j$, where $k_j = |D \setminus D_{<j}|$ is the number of terminals remaining when constructing H_j. Therefore,

$$c(H) \leq \sum_{j \in [\ell]} |D_j| \cdot O(\log^5 k_j) \frac{\mathrm{OPT}(G)}{|D \setminus D_{<j}|} \leq O(\log^5 k)\mathrm{OPT}(G) \sum_{j \in [\ell]} \frac{|D \setminus D_{<j}| - |D \setminus D_{<j+1}|}{|D \setminus D_{<j}|}.$$

The term $\sum_{j \in [\ell]} \frac{|D \setminus D_{<j}| - |D \setminus D_{<j+1}|}{|D \setminus D_{<j}|}$ is bounded by the k'th harmonic number H_k, thus $c(H)$ is at most $O(\log^6 k)\mathrm{OPT}(G)$. ∎

5 Conclusion

Several open questions arise from our work in this paper. It is unlikely that the $O(\log^6 k)$ approximation ratio that we obtained is tight. There are no known lower bounds that rule out a constant-factor approximation for DSF in planar graphs. Closing this gap is a compelling question. Second, can we establish a polylogarithmic ratio upper bound on the integrality gap of the natural cut-based LP relaxation for planar-DSF? Our techniques do not directly generalize to fractional solutions (see Remark 1.2); however, we are hopeful that other approaches may yield positive results. Another direction for future research is to extend this work and also the recent work on DST and related problems from planar graphs to any proper minor-closed family of graphs. Finally, there are several generalizations of DST and DSF that may also admit positive results in planar graphs.

Acknowledgements: We thank the anonymous reviewers for helpful suggestions and pointers to related work.

References

[1] Ajit Agrawal, Philip Klein, and R. Ravi. When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks. *SIAM Journal on Computing*, 24(3):440–456, June 1995. Publisher: Society for Industrial and Applied Mathematics. URL: https://epubs.siam.org/doi/10.1137/S0097539792236237, doi:10.1137/S0097539792236237.

[2] Mohammadhossein Bateni, Mohammadtaghi Hajiaghayi, and Dániel Marx. Approximation Schemes for Steiner Forest on Planar Graphs and Graphs of Bounded Treewidth. *J. ACM*, 58(5):21:1–21:37, October 2011. doi:10.1145/2027216.2027219.

[3] P. Berman and V. Ramaiyer. Improved Approximations for the Steiner Tree Problem. *Journal of Algorithms*, 17(3):381–408, November 1994. URL: https://www.sciencedirect.com/science/article/pii/S0196677484710418, doi:10.1006/jagm.1994.1041.

[4] 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). URL: https://www.sciencedirect.com/science/article/pii/S0890540112001484, doi:https://doi.org/10.1016/j.ic.2012.10.007.

[5] Glencora Borradaile, Philip Klein, and Claire Mathieu. An $O(nlogn)$ approximation scheme for steiner tree in planar graphs. *ACM Transactions on Algorithms (TALG)*, 5(3):1–31, 2009.

[6] Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *Journal of the ACM (JACM)*, 60(1):1–33, 2013.

[7] Gruia Calinescu and Alexander Zelikovsky. The polymatroid steiner problems. *Journal of Combinatorial Optimization*, 9:281–294, 2005.

[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, October 1999. URL: https://www.sciencedirect.com/science/article/pii/S0196677499910428, doi:10.1006/jagm.1999.1042.

[9] 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, January 2010. URL: http://epubs.siam.org/doi/10.1137/090750317, doi:10.1137/090750317.

[10] 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*, 7(2):18:1–18:17, March 2011. doi:10.1145/1921659.1921664.

[11] 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 *32nd Annual European Symposium on Algorithms (ESA 2024)*, volume 308, pages 42:1–42:19, 2024. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2024.42, doi:10.4230/LIPIcs.ESA.2024.42.

[12] 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.

[13] 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.

[14] Miroslav Chlebík and Janka Chlebíková. The steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science*, 406(3):207–214, 2008.

[15] 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*,

STOC 2022, pages 343–356, New York, NY, USA, June 2022. Association for Computing Machinery. doi:10.1145/3519935.3520049.

[16] Erik D. Demaine, Mohammadtaghi Hajiaghayi, and Philip N. Klein. Node-weighted steiner tree and group steiner tree in planar graphs. *ACM Trans. Algorithms*, 10(3), jul 2014. doi:10.1145/2601070.

[17] 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 '99, pages 750–759, New York, NY, USA, May 1999. Association for Computing Machinery. URL: https://dl.acm.org/doi/10.1145/301250.301447, doi:10.1145/301250.301447.

[18] S. E. Dreyfus and R. A. Wagner. The steiner problem in graphs. *Networks*, 1(3):195–207, 1971. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.3230010302, doi:https://doi.org/10.1002/net.3230010302.

[19] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[20] Jon Feldman and Matthias Ruhl. The Directed Steiner Network Problem is Tractable for a Constant Number of Terminals. *SIAM Journal on Computing*, 36(2):543–561, January 2006. Publisher: Society for Industrial and Applied Mathematics. URL: https://epubs.siam.org/doi/abs/10.1137/S0097539704441241, doi:10.1137/S0097539704441241.

[21] 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. URL: https://www.sciencedirect.com/science/article/pii/S0022000011000584, doi:10.1016/j.jcss.2011.05.009.

[22] Andreas Emil Feldmann and Dániel Marx. The Complexity Landscape of Fixed-Parameter Directed Steiner Network Problems. *ACM Trans. Comput. Theory*, 15(3-4):4:1–4:28, December 2023. doi:10.1145/3580376.

[23] 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. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX/RANDOM.2023.13, doi:10.4230/LIPIcs.APPROX/RANDOM.2023.13.

[24] Zachary Friggstad and Ramin Mousavi. An $O(\log k)$-Approximation for Directed Steiner Tree in Planar Graphs. In *Proceedings of ICALP*, volume 261, pages 63:1–63:14, 2023. doi:10.4230/LIPIcs.ICALP.2023.63.

[25] Naveen Garg, Goran Konjevod, and Ramamoorthi Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000.

[26] 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.

[27] 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, 2012.

[28] Michel X. Goemans and David P. Williamson. A General Approximation Technique for Constrained Forest Problems. *SIAM Journal on Computing*, 24(2):296–317, April 1995. Publisher: Society for Industrial and Applied Mathematics. URL: https://epubs.siam.org/doi/10.1137/S0097539793242618, doi:10.1137/S0097539793242618.

[29] 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.

[30] Jiong Guo, Rolf Niedermeier, and Ondřej Suchý. Parameterized Complexity of Arc-Weighted Directed Steiner Problems. *SIAM Journal on Discrete Mathematics*, 25(2):583–599, January 2011. Publisher: Society for Industrial and Applied Mathematics. URL: https://epubs.siam.org/doi/10.1137/100794560, doi:10.1137/100794560.

[31] Mohammad Taghi Hajiaghayi, Guy Kortsarz, and Mohammad R. Salavatipour. Approximating Buy-at-Bulk and Shallow-Light k-Steiner Trees. *Algorithmica*, 53(1):89–103, January 2009. doi:10.1007/s00453-007-9013-x.

[32] Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 585–594, 2003.

[33] Stefan Hougardy and Hans Jürgen Prömel. A 1.598 approximation algorithm for the Steiner problem in graphs. In *Proceedings of the tenth annual ACM-SIAM Symposium on Discrete algorithms*, SODA '99, pages 448–453, USA, January 1999. Society for Industrial and Applied Mathematics.

[34] Kamal Jain. A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. *Combinatorica*, 21(1):39–60, January 2001. doi:10.1007/s004930170004.

[35] Marek Karpinski and Alexander Zelikovsky. New Approximation Algorithms for the Steiner Tree Problems. *Journal of Combinatorial Optimization*, 1(1):47–65, March 1997. doi:10.1023/A:1009758919736.

[36] Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Embeddings of Planar Quasimetrics into Directed ℓ_1 and Polylogarithmic Approximation for Directed Sparsest-Cut, November 2021. arXiv:2111.07974 [cs]. URL: http://arxiv.org/abs/2111.07974, doi:10.48550/arXiv.2111.07974.

[37] P. Klein and R. Ravi. A Nearly Best-Possible Approximation Algorithm for Node-Weighted Steiner Trees. *Journal of Algorithms*, 19(1):104–115, July 1995. URL: `https://www.sciencedirect.com/science/article/pii/S0196677485710292`, `doi:10.1006/jagm.1995.1029`.

[38] Shi Li and Bundit Laekhanukit. Polynomial integrality gap of flow lp for directed steiner tree. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3230–3236. SIAM, 2022.

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

[40] Gabriel Robins and Alexander Zelikovsky. Tighter Bounds for Graph Steiner Tree Approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122–134, January 2005. Publisher: Society for Industrial and Applied Mathematics. URL: `https://epubs.siam.org/doi/10.1137/S0895480101393155`, `doi:10.1137/S0895480101393155`.

[41] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6):993–1024, November 2004. `doi:10.1145/1039488.1039493`.

[42] A. Z. Zelikovsky. An 11/6-approximation algorithm for the network steiner problem. *Algorithmica*, 9(5):463–470, May 1993. URL: `http://link.springer.com/10.1007/BF01187035`, `doi:10.1007/BF01187035`.

[43] Alexander Zelikovsky. A series of approximation algorithms for the acyclic directed steiner tree problem. *Algorithmica*, 18(1):99–110, 1997.

[44] Leonid Zosin and Samir Khuller. On directed steiner trees. In *Proceedings of ACM-SIAM SODA*, pages 59–63, 2002.