A New Approach for Approximating Directed Rooted Networks

Sarel Cohen[1][0000−0003−4578−1245], Lior Kamma[1][0000−0002−2380−7195], and Aikaterini Niklanovits[2][0000−0002−4911−4493]

[1] The Academic College of Tel Aviv-Yaffo, Israel {sarelco, liorkm}@mta.ac.il
[2] Hasso Plattner Institute, University of Potsdam, Germany
Aikaterini.Niklanovits@hpi.de

Abstract. We consider the *k-outconnected directed Steiner tree* problem (k-DST). Given a directed edge-weighted graph $G = (V, E, w)$, where $V = \{r\} \cup S \cup T$, and an integer k, the goal is to find a minimum cost subgraph of G in which there are k edge-disjoint rt-paths for every terminal $t \in T$. The problem is known to be NP-Hard. Furthermore, the question on whether a polynomial time, subpolynomial approximation algorithm exists for k-DST was answered negatively by Grandoni *et al.* (2018), by proving an approximation hardness of $\Omega(|T|/\log|T|)$ under NP≠ZPP.

Inspired by modern day applications, we focus on developing efficient algorithms for k-DST in graphs where terminals have out-degree 0, and furthermore constitute the vast majority in the graph. We provide the first approximation algorithm for k-DST on such graphs, in which the approximation ratio depends (primarily) on the size of S. We present a randomized algorithm that finds a solution of weight at most $O(k|S|\log|T|)$ times the optimal weight, and with high probability runs in polynomial time.

1 Introduction

Network design problems deal with finding, inside a large network G, a cheap subnetwork satisfying certain criteria. This class of problems captures a wide variety of theoretical problems, among which are the minimum spanning tree problem, minimum Steiner tree (or forest) problem or minimum k-connected subgraph in both directed and undirected graphs (see, e.g. [18, 21, 14]). In addition to its theoretical importance, network design research is of great interest for developers of practical networks such as telecommunication networks.

Our focus in this paper is the *minimum k-outconnected directed Steiner tree* problem (k-DST). An instance of k-DST consists of a directed edge-weighted graph $G = (V, E, w)$, a designated node $r \in V$ called the *root*, a subset $T \subseteq V \setminus \{r\}$ of vertices called *terminals* and an integer k. Nodes in $V \setminus (T \cup \{r\})$ are referred to as *Steiner nodes*, and the set of Steiner nodes will be denoted by S. The goal is to find a minimum-weight subgraph of G, in which for every $t \in T$ there are at least k edge-disjoint rt-paths. From a theoretical perspective, k-DST is one of the most fundamental and intensively studied problems in computer science. The most extensively studied variation of the problem is the *minimum directed Steiner tree* (DST), in which $k = 1$. This is, in part, due to the fact that many combinatorial problems are known to reduce to DST. Notable such examples among network design problems are the minimum node-weighted Steiner tree [20] in both directed and undirected graphs and the group Steiner problem [11].

Moreover, k-DST has many applications in network design, circuit layouts, and phylogenetic tree reconstruction [15] and specifically in designing *multicast* schemes [27, 30]. When designing a multicast routing scheme, the goal is to distribute information from a single source to multiple destinations, while minimizing the usage of network resources (e.g. cost of links). To ensure that the network is robust against link failures, we require that the network contains several source-destinations routes that are, in a sense, mutually independent. The connection to k-DST is straightforward. Modeling the network as a directed graph, where edges model point-to-point links, a multicast routing scheme which can survive $k - 1$ failures is a k-outconnected Steiner tree.

Networks Consisting of Mostly Terminals. Upon approaching k-DST, one might be inclined to think of T as a "small" designated subset of V. Indeed, in the majority of previous work, the approximation factors of approximation algorithms for k-DST mainly depend on the size of the terminal set T. In general, however, this need not be the case. In fact, modern day applications present networks in which the lion's share of the network consists of terminals. Consider, for example the problem of efficiently and reliably distributing critical healthcare information, including patient data and medical resources, ensuring minimal transmission costs. In particular, the root in this case is a central health database, the Steiner nodes are regional hospitals and clinics and the terminal set consists of remote clinics in rural areas, which are the majority of health sites and do not transmit any information.

From a theoretical point of view, k-DST is still an interesting problem when restricted to instances in which most of the nodes of G are terminals and have no outgoing edges. Specifically, known hardness results for the problem hold even when $|V \setminus T| = O(|V|^{1/d})$ for some constant $d \in \mathbb{N}$. Moreover, there is no evidence that the problem is easier (even to approximate) when restricted to instances where $|V \setminus T|$ is polylogarithmic in $|V|$.

Related Work. k-DST is known to be NP-hard, as it models, for example, the Steiner tree in undirected graphs as a special case [16]. However, many special cases are widely studied with positive results. For example, the case where $T = V \setminus \{r\}$, known as *minimum k-outconnected spanning subgraph problem* is proven to be polynomially time solvable [6], as was also generalized by Frank and Tardos [9] (see also [8]). Their algorithm was used as a subroutine to obtain a 2-approximation algorithm for the undirected variant and its generalization [19], which is proven to be NP-hard.

When the existence of non-terminal vertices in the graph is also considered, the complexity of k-DST is increased, and the focus hence swifts to approximation algorithms. For the case where $k = 1$ for example, Charikar *et al.* [3] presented a family of greedy approximation algorithms $\mathcal{A}_d$, indexed by an integer $d > 1$. By employing a technique introduced by Zelikovsky [31], for every $d > 1$ they showed a reduction of the instance G to a new instance G' consisting of a d-layered graph. The algorithm $\mathcal{A}_d$ returns a feasible solution (in G) of weight at most $O(d^3|T|^{1/d})$ times the optimal weight, and runs in time $|V|^{O(d)}$. A similar result has been given by Rothvoß [28], who applied the Lasserre hierarchy in order to obtain a family of algorithms giving similar approximation guarantees. This result was recently extended to the case $k = 2$ by Grandoni and Laekhanukit [13]. For every $d > 1$, they develop a polynomial time approximation algorithm of ratio $\tilde{O}(d^3|T|^{1/d})$ and running time $O(|V|^{O(d)})$. For the undirected case of the problem when restricting the value of k to be 1, Byrka *et al.* provided a 1.39-approximation algorithm [1].

Moreover, the *directed Steiner tree* problem (i.e. $k = 1$), is solved by a trivial algorithm in time $2^{|S|} \cdot |V|^{O(1)}$, thus if S is "small" (i.e. $|S| \leq O(\log |V|)$) the problem is solvable in polynomial time. On the other hand, the Dreyfus–Wagner algorithm [5] solves the problem in time $O(3^{|T|} \cdot |V|^2)$, thus showing that it can also be solved in polynomial time if the number of terminals is "small". It is hence natural to consider developing algorithms for k-DST whose running time or approximation ratio is based on the amount of terminals. Such approximation algorithms for k-DST are developed only for the special cases where G is an L-layered graph, by Laekhanukit who gave a $\mathcal{O}(k^L Llog|T|)$-approximation algorithm [23], and for the case where G is a quasi-bipartite graph for which Chan *et al.* gave a polynomial time $\mathcal{O}(\log |T| \log k)$-approximation algorithm [2].

When it comes to hardness of k-DST much more results have been obtained. Cheriyan *et al.* [4] proved it to be at least as hard to approximate as the label cover problem. Therefore there is no $2^{\log^{1-\varepsilon} |V|}$-approximation for k-DST for any fixed $\varepsilon > 0$ unless $NP \subseteq DTIME(n^{polylog(n)})$. Laekhanukit [22] showed that unless $NP = ZPP$, the k-DST problem admits no $k^{1/4-\varepsilon}$-approximation for any $\varepsilon > 0$. We note that this hardness result holds even when restricted to instances in which $|V \setminus T| = O(|V|^{1/d})$ for some constant $d \in \mathbb{N}$ via a simple polynomial reduction. Another polynomial reduction of k-DST that provides us with a lower bound for approximating it is the one to the *set cover* problem. Nelson [25] showed that for every $c < 0.5$, set cover cannot be approximated in polynomial time within $2^{\log^{1-\delta_c(m)}(m)}$, unless SAT on m variables can be solved in time $2^{O(2^{\log^{1-\delta_c(m)}(m)})}$, where $\delta_c(m) = 1 - \frac{1}{(\log\log m)^c}$. Through a simple polynomial time reduction we are able to obtain a lower bound for approximating k-DST (in fact, even DST) in terms of $|S|$. In particular we conclude that k-DST cannot be approximated in polynomial time to within $2^{\log^{1-\delta_c(|S|)}(|S|)}$, unless SAT on m variables can be solved in time $2^{O(2^{\log^{1-\delta_c(m)}(m)})}$. One of the most recent results is the one from Liao *et al.* [24] where the improved the prreviously known lower approximation hardness to $\Omega(|T|/\log|T|))$ under NP$\neq$ ZPP, $\Omega(2^{k/2}/k)$ under NP$\neq$ ZPP and $\Omega(k/L)^{L/4}$ on L-layered graphs for $L \leq O(\log n)$.

1.1 Our Results

Let $G = (\{r\} \cup S \cup T, E, w)$, $k \in \mathbb{N}$ be an instance of k-DST. Assume in addition that every edge $uv \in E$ satisfies that $u \notin T$. The main contribution of this paper is providing the first approximation algorithm for k-DST on graphs where the terminals do not transmit any information. Our algorithm is a randomized Las Vegas algorithm, which runs, with high probability, in polynomial time.

Theorem 1. *There is a randomized $O(k|S| \log |T|)$-approximation algorithm for k-DST when the outdegree of the vertices in T is 0, that runs in polynomial time with high probability.*

This approximation guarantee depends primarily on the size of S, and up to a logarithmic factor is independent of the size of T. If $|S| = O(|T|^{1/d})$ for some constant $d > 1$, the approximation ratio of our algorithm is comparable with that of Charikar *et al.* [3]. Moreover, if $|S|$ is polylogarithmic in $|T|$, we achieve a polylogarithmic approximation ratio (in terms of $|V|$) in polynomial time.

Overview and Techniques. Throughout this paper let $G = (\{r\} \cup S \cup T, E, w), k$ be an instance of k-DST, where $V = \{r\} \cup S \cup T$, and $\mathcal{U} := \{U \subseteq V \setminus \{r\} : U \cap T \neq \emptyset\}$. An $e = xy \in E$ *covers* $U \in \mathcal{U}$ if e enters U, i.e. $x \notin U$ and $y \in U$. Given a set $F \subseteq E$, denote by $\varrho_F(U) := \{xy \in F : x \notin U,\ y \in U\}$ the set of edges in F covering U. By the minimum-cut maximum-flow theorem, a set $F \subseteq E$ is a feasible solution for k-DST if and only if every $U \in \mathcal{U}$ is covered by at least k edges, i.e. $|\varrho_F(U)| \geq k$.

Connectivity Augmentation. An intermediate problem between DST and k-DST is the one of increasing the rooted outconnectivity of a graph by one (connectivity augmentation problem). Formally, let $\ell \in \mathbb{N}$. Given a graph $G = (V, E, w)$, a root $r \in V$, a set of terminals $T \subseteq V \setminus \{r\}$, and a set $E_\ell \subseteq E$ of edges such that in the subgraph (V, E_ℓ) of G there are ℓ edge-disjoint rt-paths for every $t \in T$, we seek a minimum cost set $F \subseteq E \setminus E_\ell$ such that in $(V, E_\ell \cup F)$ there are $\ell + 1$ edge disjoint rt-paths for every $t \in T$. An additional result presented in this paper is the following.

Theorem 2. *There is a randomized approximation algorithm for the connectivity augmentation problem when the outdegree of the vertices in T is 0, that constructs an $O(|S| \log |T|)$-approximate solution. Also, with probability at least $1 - \frac{\log |T|}{2^{|E|}}$ the algorithm runs in polynomial time.*

The connection to k-DST is quite clear. Starting with an empty set of edges, our approximation algorithm iteratively finds small-weight solutions to the connectivity augmentation problem k times, and produces edge sets $E_1, \ldots, E_k \subseteq E$ whose union is a small-weight feasible solution to k-DST, incurring an additional factor of k in the approximation ratio. We note that very often when solving connectivity problems by repeatedly invoking a connectivity augmentation mechanism, it is possible to reduce the additional factor to the approximation ratio from k to $\log k$, using a standard linear-programming scaling technique introduced by Goemans *et al.* [12]. The algorithm presented in this paper for the augmentation problem, however, is invoked on an auxiliary graph, rather than on G. This auxiliary graph, and thus the induced linear program, change between invocations, and hence it is not clear whether the scaling technique is applicable.

Given an instance to the connectivity augmentation problem, and applying the minimum-cut maximum-flow theorem once more, we deduce that a set $F \subseteq E \setminus E_\ell$ is a feasible solution to the problem if and only if $|\varrho_{E_\ell \cup F}(U)| \geq \ell + 1$ for every $U \in \mathcal{U}$. From our assumptions on E_ℓ, we have $|\varrho_{E_\ell}(U)| \geq \ell$ for every $U \in \mathcal{U}$. In addition $F \cap E_\ell = \emptyset$, and therefore F is a feasible solution if and only if $\varrho_F(U) \neq \emptyset$ for all $U \in \mathcal{U}_\ell$, where $\mathcal{U}_\ell := \{U \in \mathcal{U} : |\varrho_{E_\ell}(U)| = \ell\}$. Hence, the augmentation problem is formulated as the following integer program.

$$\min \textstyle\sum_{e \in E} w_e x_e \qquad s.t. \qquad \begin{array}{ll} \sum_{e \in \varrho_{E \setminus E_\ell}(U)} x_e \geq 1 & \forall U \in \mathcal{U}_\ell, \\ x_e \in \{0,1\} & \forall e \in E \end{array} \tag{IP}$$

Implicit Hitting Set. In the *hitting set* problem, we are given a ground set E, with weights $\{w_e\}_{e \in E}$, and a collection $\mathcal{S} \subseteq 2^E$ of subsets of E. The goal is to find a minimum weight subset $F \subseteq E$ such that $F \cap U \neq \emptyset$ for all $U \in \mathcal{S}$. The problem is known to be NP-hard [16], and can be approximated by an $O(\log |\mathcal{S}|)$-factor via a greedy algorithm. Karp *et al.* [17] observed that many combinatorial optimization problems possess the following property. Given an instance Π, we construct in polynomial time an instance $E, w, \mathcal{S}$ to the hitting set problem, where E, w are given explicitly, and $\mathcal{S}$, whose size might be exponential in the size of Π, is given implicitly by a membership oracle. Karp *et al.* named this setting the *implicit hitting set* problem, and observed that the well-known greedy algorithm for the hitting-set problem cannot be used to approximate this type of problems, as it examines all sets in $\mathcal{S}$.

We show that under reasonable assumptions we still achieve a randomized polynomial time $O(\log |\mathcal{S}|)$-approximation algorithm. Specifically, we slightly modify the well-known randomized rounding algorithm for hitting set, and devise a $O(\log |\mathcal{S}|)$-approximation algorithm that, with very high probability, say $\geq 1 - 2^{-|E|}$, runs in time polynomial to $|E|$. Details are deferred to Appendix A. The connectivity augmentation problem can be viewed as an implicit hitting set instance, where $\mathcal{S} := \{\varrho_{E \setminus E_\ell}(U) : U \in \mathcal{U}_\ell\}$. However, even for $\ell = 0$ we get $\log |\mathcal{S}| = \Omega(|V|)$. Achieving a $|T|$-approximation for the problem is immediate (increase the rt-connectivity by one for each $t \in T$ separately), hence we seek to exploit its structure, and specifically the structure of $\mathcal{U}_\ell$ to improve our guarantees.

A set-family $\mathcal{G} \subseteq 2^V$ is called *intersecting*, if when $X, Y \in \mathcal{G}$ and $X \cap Y \neq \emptyset$, then $X \cap Y, X \cup Y \in \mathcal{G}$. Frank [10] observed that if $T = V \setminus \{r\}$, the family $\mathcal{U}_\ell$ is intersecting, and also showed that the problem of covering intersecting set-families by directed edges can be solved optimally efficiently. In general, however, $\mathcal{U}_\ell$ is not intersecting. In a following paper [7] Frank coined the refined notion of T-Intersecting Families.

Definition 1. *A set family $\mathcal{G} \subseteq 2^V$ is called T-*intersecting *if for every $X \in \mathcal{G}$, $X \cap T \neq \emptyset$; and for every $X, Y \in \mathcal{G}$, if $X \cap Y \cap T \neq \emptyset$ then $X \cap Y, X \cup Y \in \mathcal{G}$.*

The proof of Proposition 1 is moved to the Appendix due to lack of space.

Proposition 1. *$\mathcal{U}_\ell$ is T-intersecting.*

Frank considered the problem of covering T-intersecting families, and showed that if every edge of G enters T, then a T-intersecting family can be covered efficiently using the primal dual approach. For arbitrary graphs, however, the problem of optimally covering a T-intersecting family of sets is NP-hard as it models the directed Steiner tree problem as a special case. By taking advantage of several structural properties of T-intersecting families, we present a mechanism for constructing a small-weight cover of $\mathcal{U}_\ell$. The mechanism is presented in detail in Section 3. Loosely speaking, we iteratively prune $\mathcal{U}_\ell$, covering small sub-families of $\mathcal{U}_\ell$ one at a time. Since the approximation ratio of the hitting set randomized algorithm depends on the size of the family to be covered, the problem of covering a small sub-family of $\mathcal{U}_\ell$ can be better approximated. We show that by pruning $\mathcal{U}_\ell$ carefully, the process ends after a small number of iterations, thus constructing a feasible solution with better approximation guarantees.

Strict Cores of T-Intersecting Families. In [26] Nutov studied rooted connectivity problems and defined the notion of a *core* of a set family. Given a set family $\mathcal{G} \subseteq 2^V$, a set $X \in \mathcal{G}$ is a core of $\mathcal{G}$ if it contains exactly one inclusion-minimal set of $\mathcal{G}$. Nutov gave an algorithm for covering so-called bi-uncrossable set families by iteratively covering the sub-families of cores. Formally, the iterative process is as follows. Given a set family $\mathcal{G}$, find a small-weight edge set $F_1 \subseteq E$ that covers all cores of $\mathcal{G}$. Denote by $\mathcal{G}^{F_1}$ the family of all sets in $\mathcal{G}$ not covered by F_1. Continue iteratively by finding $F_2, \ldots, F_t$ such that for every $j \in [t]$, F_j covers all cores of $\mathcal{G}^{F_1 \cup \ldots \cup F_{j-1}}$. Nutov additionally bounded the number of iterations needed to cover bi-uncrossable set families, when iteratively covering the sub-families of cores. In Section 3 we show that by iteratively covering the sub-family of cores of the T-intersecting family $\mathcal{U}_\ell$, we can guarantee that no more than $\log |T|$ iterations are needed. However we also show, the sub-family of cores can be as large as the entire $\mathcal{U}_\ell$, and hence our approximation guarantees are still too large.

Therefore, in this paper we present the new notion of *strict cores.* A strict core of $\mathcal{U}_\ell$ is a set $X \in \mathcal{U}_\ell$ that contains exactly one inclusion-minimal set $C \in \mathcal{U}_\ell$, and satisfies $X \cap T = C \cap T$. We show that the sub-family of strict cores of $\mathcal{U}_\ell$ is significantly smaller than the sub-family of cores, and give a polynomial time approximation algorithm for covering the sub-family of strict cores. As we show in Section 3.1 we can make no guarantee, however, on the number of iterations needed to iteratively cover the sub-family of strict cores. The main technical crux of this paper is therefore the construction of a new auxiliary graph in which every edge cover to the family of strict cores also covers the family of cores.

2 Approximating k-Outconnected Steiner Tree

We begin the proof of Theorem 1 by showing that it is implied by Theorem 2. To this end, let $G = (\{r\} \cup S \cup T, E, w)$ and k be an instance of k-DST. Denote by $\mathrm{opt}(G)$ the value of an optimal solution, and let $\mathcal{U} = \{U \subseteq V \setminus \{r\} : U \cap T \neq \emptyset\}$. We present an iterative algorithm that finds a feasible solution for k-DST, by iteratively increasing the connectivity between the root and the terminals. The algorithm performs k iterations numbered $0, 1, \ldots, k-1$. For every $0 \leq \ell \leq k-1$, in the beginning of the ℓth iteration, the algorithm holds a set $E_\ell \subseteq E$ of edges, starting with $E_0 = \emptyset$, such that in (V, E_ℓ), there are at least ℓ edge-disjoint rt-paths for every $t \in T$. During the ℓth iteration, the algorithm invokes the $O(|S| \log |T|)$-approximation algorithm whose existence is implied by Theorem 2, to find an approximate solution $F \subseteq E \setminus E_\ell$ to the connectivity augmentation instance given by G, E_ℓ. The algorithm is described in detail as Algorithm 1.

```
1: $E_0 = \emptyset$.
2: for $\ell = 0, 1, \ldots, k-1$ do
3:     let $F \subseteq E \setminus E_\ell$ be an $O(|S| \log |T|)$-approximate solution to the connectivity
       augmentation problem on $G, E_\ell$ (apply the algorithm in Theorem 2).
4:     let $E_{\ell+1} = E_\ell \cup F$.
5: return $E_k$.
```

Algorithm 1: Approximation Algorithm for k-DST

Through a simple induction we are able to see that for every $t \in T$, there are k edge-disjoint rt-paths in (V, E_k). This also implies that the edge set constructed by Algorithm 1 is a feasible solution to k-DST. The following lemma shows that the weight of this set is at most $O(k|S| \log |T|)$ times the optimum.

Lemma 1. *$w(E_k) \leq O(k|S| \log |T|) \cdot \mathrm{opt}(G)$.*

Proof. Let $0 \leq \ell \leq k-1$. Consider an optimal solution $F^* \subseteq E$ to the k-DST problem on G. Then for every $U \in \mathcal{U}$, $|\varrho_{F^*}(U)| \geq k$. Let $U \in \mathcal{U}$ be such that $|\varrho_{E_\ell}(U)| = \ell < k$. Then $|\varrho_{F^* \setminus E_\ell}(U)| \geq k - \ell \geq 1$. Therefore $F^* \setminus E_\ell$ is a feasible solution for the augmentation problem defined by G, E_ℓ. We conclude that the set F constructed in line 3 satisfies $w(F) \leq O(|S| \log |T|)\, \mathrm{opt}(G)$. The lemma follows.

To conclude the proof of Theorem 1 it remains to show that the algorithm runs in polynomial time with high probability. Theorem 2 guarantees that line 3 always returns an approximate solution, and it runs in polynomial time with probability at least $1 - \frac{\log |T|}{2^{|E|}}$. Since Algorithm 1 performs k iterations, we get by a union bound that Algorithm 1 runs in polynomial time with probability at least $1 - \frac{k \log |T|}{2^{|E|}}$, and since $k \log |T| \leq |E|^2$, this probability is very high. The remainder of the paper is devoted to the proof of Theorem 2.

3 Increasing Rooted Connectivity By One

In this section, we prove Theorem 2 by presenting an $O(|S|\log|T|)$-approximation algorithm for the connectivity augmentation problem. Let $G = (\{r\} \cup S \cup T, E, w)$ be the instance graph, and $E_\ell \subseteq E$ be a set of edges such that in (V, E_ℓ) there are ℓ edge-disjoint rt-paths for every $t \in T$. Let $\mathcal{F} := \{U \subseteq V \setminus \{r\} : U \cap T \neq \emptyset \textit{ and } |\varrho_{E_\ell}(U)| = \ell\}$, then Proposition 1 implies $\mathcal{F}$ is T-intersecting. By the min-cut max-flow theorem, a set $F \subseteq E \setminus E_\ell$ is a feasible solution for the augmentation problem if and only if $\varrho_F(U) \neq \emptyset$ for every $U \in \mathcal{F}$. The linear program below can be viewed as a fractional relaxation of the augmentation problem.

$$\begin{array}{llll} min \sum_{e\in E} w_e x_e & s.t. & \sum_{e\in\varrho_{E\setminus E_\ell}(U)} x_e \geq 1 & \forall U \in \mathcal{F}, \\ & & x_e \geq 0 \qquad \forall e \in E & \end{array} . \tag{LP}$$

Fix some $x \in \mathbb{R}_+^E$. For every $e \in E$, we think of x_e as the capacity of e in G. Define $\hat{x} \in \mathbb{R}_+^E$ by $\hat{x}_e = 1$ for all $e \in E_\ell$ and $\hat{x}_e = x_e$ otherwise. Then x is a feasible solution for (LP) if and only if the capacity of a minimum rt-cut in $(G, \hat{x})$ is at least $\ell + 1$ for every $t \in T$. Therefore, the feasibility of x can be verified in time polynomial in the size of G. Moreover, if x is not feasible, there is some $t \in T$ such that in $(G, \hat{x})$ there is an rt-cut of capacity less than $\ell + 1$. Such a cut $U \subseteq V$ is a violated constraint and is found in polynomial time. The program (LP) can therefore be solved efficiently using a separation oracle. With high probability we thus efficiently obtain an $O(\log|\mathcal{F}|)$-approximate solution for the connectivity augmentation problem. However, in general $\log|\mathcal{F}| = \Omega(|V|)$, hence the approximation factor might be too large, since obtaining approximation of factor $|T|$ times the optimal value is trivial (increase the rt-connectivity for every $t \in T$ separately). Note that we did not use any properties of $\mathcal{F}$, other than finding an efficient separation oracle for (LP). Specifically, we did not use the structural properties of T-intersecting families.

In what follows, we present a mechanism for covering the T-intersecting family $\mathcal{F}$ by edges. Loosely speaking, we iteratively prune $\mathcal{F}$ by covering small sub-families. For each such sub-family we efficiently find a small weight cover. By choosing the sub-families carefully, the process ends after a small number of iterations. More formally, we look for sub-families $\mathcal{F}_1, \ldots, \mathcal{F}_t \subseteq \mathcal{F}$ such that (i) if $F \subseteq E \setminus E_\ell$ covers $\mathcal{F}_1, \ldots, \mathcal{F}_t$ then F covers $\mathcal{F}$; (ii) for every $j \in [t-1]$, given a cover $F \subseteq E \setminus E_\ell$ for $\mathcal{F}_1, \ldots, \mathcal{F}_j$, we can efficiently find a small-weight cover $F' \subseteq E \setminus E_\ell$ for $\mathcal{F}_{j+1}$; and (iii) t is small.

We start by exhibiting several fundamental properties of $\mathcal{F}$. Given a set $F \subseteq E \setminus E_\ell$, a set $X \in \mathcal{F}$ is called *F-tight* if F does not cover X, i.e. $\varrho_F(X) = \emptyset$. Denote by $\mathcal{F}^F$ the family of F-tight elements of $\mathcal{F}$. A crucial property of $\mathcal{F}$ is that the structural property of being T-intersecting is, in a sense, hereditary. In particular it is easy to prove that $\mathcal{F}^F$ is T-intersecting as follows. Let $X, Y \in \mathcal{F}^F$ be such that $X \cap Y \cap T \neq \emptyset$. Then $X \cap Y, X \cup Y \in \mathcal{F}$. Furthermore, by submodularity of the cut function, $0 = |\varrho_F(X)| + |\varrho_F(Y)| \geq |\varrho_F(X \cup Y)| + |\varrho_F(X \cap Y)| \geq 0$. Therefore $|\varrho_F(X \cup Y)| = |\varrho_F(X \cap Y)| = 0$, and thus $X \cap Y$ and $X \cup Y$ are F-tight elements of $\mathcal{F}$, that is $X \cup Y, X \cap Y \in \mathcal{F}^F$.

The following definition, and the two lemmas that follow it are inspired by a similar observation by Nutov [26] for problems regarding undirected graphs. The proofs of Lemmas 2,3 are moved to the Appendix due to space restrictions.

Definition 2. *Let $\mathcal{M}(\mathcal{F}^F)$ denote the family of inclusion-minimal elements of $\mathcal{F}^F$. A set $X \in \mathcal{F}^F$ is called an $\mathcal{F}^F$-*core *if it contains exactly one inclusion-minimal element of $\mathcal{F}^F$.*

Lemma 2. *Let $X \in \mathcal{F}^F$. Then for every $C \in \mathcal{M}(\mathcal{F}^F)$, either $C \cap X \cap T = \emptyset$ or $C \subseteq X$. In particular, for every distinct $C, D \in \mathcal{M}(\mathcal{F}^F)$, $C \cap D \cap T = \emptyset$.*

Lemma 3. *If $F' \subseteq E \setminus E_\ell$ covers all $\mathcal{F}^F$-cores. Then $|\mathcal{M}(\mathcal{F}^{F \cup F'})| \leq \frac{1}{2}|\mathcal{M}(\mathcal{F}^F)|$.*

Lemma 3 implies that by iteratively covering the sub-family of tight cores, we cover $\mathcal{F}$ after at most $\log|\mathcal{M}(\mathcal{F})|$ iterations. Since $|\mathcal{M}(\mathcal{F})| \leq |T|$, we conclude that the process terminates after at most $\log|T|$ iterations. The next section constitutes the technical crux in the proof of Theorem 2.

3.1 Covering Cores Cheaply and Efficiently

In this section we conclude proof of Theorem 2 by giving an $O(|S|)$-approximation algorithm for the problem of covering the family of $\mathcal{F}^F$-cores. A naïve approach suggested by the preceding discussion is to simply apply the implicit hitting set approximation algorithm to cover the family of $\mathcal{F}^F$-cores. However, the sub-family of $\mathcal{F}^F$-cores can be almost as large as the entire $\mathcal{F}^F$. Hence the approximation factor might be as large as $\Omega(|V|)$. We therefore refine the definition of cores. This is where we diverge from [26], and from all previous work.

Definition 3. *A set $X \in \mathcal{F}^F$ is called a* strict $\mathcal{F}^F$-core *if there exists $C \in \mathcal{M}(\mathcal{F}^F)$ such that $X \cap T = C \cap T$, that is, X and C have the same set of terminals.*

One can easily verify, using Lemma 2 that every strict $\mathcal{F}^F$-core contains exactly one inclusion-minimal set of $\mathcal{F}^F$. Thus every strict $\mathcal{F}^F$-core is also an $\mathcal{F}^F$-core. As opposed to the family of tight cores, the family of strict cores is significantly small.

Lemma 4. *The number of strict $\mathcal{F}^F$-cores is at most $2^{|S|} \cdot |T|$.*

Proof. Let X be a strict $\mathcal{F}^F$-core. Then there is a unique $C \in \mathcal{M}(\mathcal{F}^F)$ such that $C \subseteq X$ and $C \cap T = X \cap T$. Therefore $X \setminus C \subseteq S$, and hence the family of strict $\mathcal{F}^F$-cores is contained in $\{C \cup X' : C \in \mathcal{M}(\mathcal{F}^F),\ X' \in 2^S\}$.

By Lemma 4, if we can model the problem of covering all strict $\mathcal{F}^F$-cores as an implicit hitting set problem, the approximation factor of the randomized rounding algorithm is reduced to $O(|S| + \log|T|) = O(|S|)$. However, if $F' \subseteq E \setminus E_\ell$ covers all strict cores, it does not necessarily cover all cores. Nevertheless, not all is lost.

In a sense, the essence of the inadequacy of covering only strict cores lies in the following scenario. Consider an edge e leaving a covered core X and entering a non-covered strict core C covers the strict core, but does not reduce the number of non-covered minimal cores. Covering all strict cores with such edges attains no advancement to the algorithm. We thus turn to construct an auxiliary graph, in which this scenario cannot occur.

More formally, given G, E_ℓ and F, let $\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$ denote the weight of an optimal solution for the augmentation problem. We show that we can construct a new graph G^F on the same vertex set V, in which (i) we can find in G^F an edge set $A \subseteq E(G^F)$ covering all $\mathcal{F}^F$-cores such that $w(A) \leq O(|S|)\,\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$; and (ii) given a cover $A \subseteq E(G^F)$ for the family of $\mathcal{F}^F$-cores, we can construct a cover $F' \subseteq E \setminus E_\ell$ of the $\mathcal{F}^F$-cores in G such that $w(F') \leq w(A)$. Moreover, the construction of G^F, as well as (i) and (ii), can be done in polynomial time with high probability. Note that unlike simple reductions commonly used in dealing with the directed Steiner tree problem (e.g. reducing the instance to an acyclic graph or a layered graph), the construction we present preserves the vertex set intact, and specifically does not change $|S|$.

Following the discussion above, we want the weights to satisfy the triangle inequality, and therefore we first define a special form of metric completion of G. In classical literature, for every $u, v \in V$, the weight of the edge uv in the metric completion of a weighted graph G is defined to be the length of a shortest uv-path in G. By the minimum-cut maximum-flow theorem, for every $u, v \in V$, a shortest uv-path in G is a minimum-weight edge-cover of the set $\{U \subseteq V \setminus \{u\} : u \in U\}$. Therefore an analogous way of viewing the definition of a metric completion is the following. For every $u, v \in V$, the weight of the edge uv in the metric completion of G is defined to be the minimum weight of an edge cover for the set $\{U \subseteq V \setminus \{u\} : u \in U\}$. In our setting not all such subsets $U \subseteq V \setminus \{u\}$ need to be covered. We therefore wish to refine the classical notion of a metric completion of G. Specifically, if every $X \in \mathcal{F}^F$ such that $v \in X$ satisfies $u \in X$, then in a sense we do not need to cover any uv-cut, since all *relevant* cuts (i.e. those belonging to $\mathcal{F}^F$) have been covered. We can therefore set the weight of the edge uv to zero.

Formally we define a graph $G^0 = (V, (E \setminus E_\ell) \cup E^0, w^0)$ as $E^0 = \{uv :$ for every $X \in \mathcal{F}^F$, if $v \in X$ then $u \in X\}$ and constructed through the following procedure. For every $e \in (E \setminus E_\ell) \cup E^0$, we define $w^0_e = 0$ if $e \in F \cup E^0$, and $w^0_e = w_e$ otherwise. Next, let G^1 be the standard metric completion of G^0. That is, G^1 is the complete directed graph on V, where w^1_{uv} is the length of a shortest uv-path in G^0. The weight assignment w^1 satisfies the triangle inequality, and is therefore an asymmetric metric.

Next, following the discussion above, we construct G^F by removing from G^1 edges, which we can assert will not belong to any inclusion-minimal solution. Formally, G^F is the subgraph of G^1 constructed as follows. For every $s \in \bigcup_{C \in \mathcal{M}(\mathcal{F}^F)} (C \cap T)$ and for every $u \in V \setminus \left(\bigcup_{C \in \mathcal{M}(\mathcal{F}^F)} (C \cap T)\right)$, if every $X \in \mathcal{F}^F$ that contains s also contains u, remove from G^1 all edges outgoing from u. The proof of the claim that G^F can be constructed in polynomial time given G, E_ℓ, F is deferred to Appendix C.

Lemma 5. *For every $A \subseteq E(G^F)$, if A covers all $\mathcal{F}^F$-cores, we can construct in polynomial time an edge set $F' \subseteq E \setminus E_\ell$ that covers all $\mathcal{F}^F$-cores such that $w(F') \leq w^1(A)$.*

Proof. Let $A \subseteq E(G^F)$ be an $\mathcal{F}^F$-cores cover in G^F, $e = uv \in A$, and $P_e \subseteq (E \setminus E_\ell) \cup E^0$ be a shortest uv-path in G^0. Then $w^1_e = w^0(P_e) = \sum_{e' \in P_e} w^0_{e'} = \sum_{e' \in P_e \cap (E \setminus (E_\ell \cup F))} w_{e'}$. Let $X \in \mathcal{F}^F$ be an $\mathcal{F}^F$-core covered by e. Then $u \notin X$ and $v \in X$. Hence there is an edge $e' = u'v' \in P_e$ that covers X. Since $X \in \mathcal{F}^F$ and $v' \in X$ and $u' \notin X$, it follows that $e' = u'v' \notin F \cup E^0$, and hence $e' \in E \setminus (E_\ell \cup F)$. We conclude that $F' := \bigcup_{e \in A} (P_e \cap (E \setminus (E_\ell \cup F))$ covers all $\mathcal{F}^F$-cores, and $w(F') = \sum_{e \in F'} w_e \leq \sum_{e \in A} \sum_{e' \in P_e \cap (E \setminus (E_\ell \cup F))} w_{e'} = \sum_{e \in A} w^1_e = w^1(A)$.

Due to Lemma 5, in order to find a low-cost cover $F' \subseteq E \setminus E_\ell$ for all $\mathcal{F}^F$-cores it suffices to find a low-cost cover $A \subseteq E(G^F)$. The following two lemmas show that such a cover exists, and that it is enough for A to cover the strict $\mathcal{F}^F$-cores.

Lemma 6. *There exists an edge set $A \subseteq E(G^F)$ such that A covers all $\mathcal{F}^F$-cores and $w^1(A) \leq \text{opt}_{\text{aug}}(G, E_\ell)$.*

Proof. For every set $E' \subseteq E \setminus E_\ell$, from our construction we get that $w^1(E') \leq w(E')$, and therefore $\text{opt}_{\text{aug}}(G^1, E_\ell) \leq \text{opt}_{\text{aug}}(G, E_\ell)$. Hence, there is an edge set $A' \subseteq E(G^1)$ such that in the subgraph $(V, E_\ell \cup A')$ there are $\ell + 1$ edge-disjoint rt-paths for all $t \in T$, and such that $w^1(A') \leq \text{opt}_{\text{aug}}(G, E_\ell)$. We show that there is a set $A \subseteq E(G^F)$ such that in the subgraph $(V, E_\ell \cup A)$ there are $\ell + 1$ edge-disjoint rt-paths for all $t \in T$ and $w^1(A) \leq w^1(A')$. Clearly, such A covers all strict $\mathcal{F}^F$-cores in G^F and $w^1(A) \leq \text{opt}_{\text{aug}}(G, E_\ell)$. We continue by induction on $|A' \setminus E(G^F)|$. If $|A' \setminus E(G^F)| = 0$, take $A = A'$, and the result follows. Otherwise, let $uv \in A' \setminus E(G^F)$. Then $u \in V \setminus \left(\bigcup_{C \in \mathcal{M}(\mathcal{F}^F)} (C \cap T)\right)$ and there is $s \in \bigcup_{C \in \mathcal{M}(\mathcal{F}^F)} (C \cap T)$ such that every $Z \in \mathcal{F}^F$ containing u also contains s. By the definition of A', every set $Z \in \mathcal{F}^F$ containing s is covered by A'. Let $A'' = (A' \setminus \{uv\}) \bigcup \{sv\}$. By triangle inequality, $w^1_{sv} \leq w^1_{su} + w^1_{uv}$. Since $su \in E^0$, $w^1_{su} = 0$ we get that $w^1_{sv} \leq w^1_{uv}$. Hence, $w^1(A'') \leq w^1(A')$. Since $s \in \bigcup_{C \in \mathcal{M}(\mathcal{F}^F)} (C \cap T)$, then $sv \in E(G^F)$, we get $|A'' \setminus E(G^F)| = |A' \setminus E(G^F)| - 1$.

Lemma 7. *For every $A \subseteq E(G^F)$, A covers all $\mathcal{F}^F$-cores if and only if A covers all strict $\mathcal{F}^F$-cores.*

Proof. Let $A \subseteq E(G^F)$. Clearly, if A covers all $\mathcal{F}^F$-cores, then it covers all strict $\mathcal{F}^F$-cores. Assume, therefore, that A covers all strict $\mathcal{F}^F$-cores. Let $X \in \mathcal{F}^F$ be an $\mathcal{F}^F$-core that is not a strict core. There is a unique $C \in \mathcal{M}(\mathcal{F}^F)$ such that $C \subseteq X$. Let $X' = X \setminus (T \setminus C)$. First note that since for every $u \in T \setminus C$, no edge leaves u, then by removing from X all terminals in $T \setminus C$, no new edges enter X'. Therefore $X' \in \mathcal{F}$. To see that $X' \in \mathcal{F}^F$, let $xy \in F$ be some edge, and assume $y \in X' \subseteq X$. Assume $y \in C$. Since $C \in \mathcal{F}^F$, then $x \in C \subseteq X'$. Otherwise, since $x \notin T$, it follows that $x \notin X \setminus X'$. Since $X \in \mathcal{F}^F$, it follows that $x \in X$, and thus $x \in X'$. Therefore xy does not cover X', and thus $X' \in \mathcal{F}^F$. Next we note that $X' \cap T = C \cap T$, and thus X' is a strict core.

By assumption on A, there is an edge $uv \in A$ such that $v \in X' \subseteq X$ and $u \notin X'$. If $u \in X$, then $u \in T$. Therefore $u \notin X$, and thus X is covered by A.

The following lemma shows that in G^F we can find with high probability an approximate cover for the set of $\mathcal{F}^F$-cores.

Lemma 8. *There is a randomized algorithm that finds a set $A \subseteq E(G^F)$ such that A covers all $\mathcal{F}^F$-cores and $w^1(A) \leq O(|S|)\,\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$. Moreover, with probability at least $1 - 2^{-|E|}$ the algorithm runs in polynomial time.*

The proof of Lemma 8 is technically involved, however the main idea is quite straightforward. First, note that by Lemma 7, it suffices to show an algorithm that finds an approximate cover for the set of strict $\mathcal{F}^F$-cores. We show that a linear program relaxation of the latter has a polynomial time separation oracle, hence the randomized rounding algorithm for the hitting set finds such a cover in polynomial time with high probability. The proof is given in detail in Appendix C.

We are now ready to prove Theorem 2, by describing an $O(|S| \log |T|)$-approximation algorithm for the augmentation problem. The algorithm iteratively prunes $\mathcal{F}$ by covering the set of tight cores in every iteration. Starting with $F = \emptyset$, and during every iteration until F covers $\mathcal{F}$, we add to F a set $F' \subseteq E \setminus E_\ell$ of weight at most $O(|S|)\,\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$ that covers the family of $\mathcal{F}^F$-cores. The mechanism is presented in detail as Algorithm 2.

```
1: $j \leftarrow 1$
2: $F_j \leftarrow \emptyset$
3: **while** $\mathcal{F}^F \neq \emptyset$ **do**
4:     *(implicitly)* let $\mathcal{F}_j$ be the set of $\mathcal{F}^F$-cores.
5:     let $F' \subseteq E \setminus E_\ell$ be an edge set covering $\mathcal{F}_j$ such that $w(F') \leq O(|S|)\,\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$.

6:     $F \leftarrow F \cup F'$.
7:     $j \leftarrow j + 1$.
8: **return** $F$.
```

Algorithm 2: $O(|S| \log |T|)$-Approximation Algorithm for the Connectivity Augmentation Problem

Proof (of Theorem 2). We first show that each iteration is executed in polynomial time with probability at least $1 - 2^{-|E|}$. Given a set $F \subseteq E \setminus E_\ell$, we efficiently (deterministically) check whether $\mathcal{F}^F = \emptyset$, by verifying that for every $t \in T$, there are $\ell+1$ edge-disjoint rt-paths in $(V, E_\ell \cup F)$. To see that line 5 runs in polynomial time with high probability, first note that given G, E_ℓ, F, we can construct G^F in polynomial time. By Lemma 8, there is a randomized algorithm that finds a set $A \subseteq E(G^F)$ such that A covers all $\mathcal{F}^F$-cores and $w^1(A) \leq O(|S|)\,\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$. Moreover, with probability at least $1-2^{-|E|}$ the algorithm runs in polynomial time. Lemma 5 then guarantees that we can construct from A an edge set $F' \subseteq E \setminus E_\ell$ that covers all $\mathcal{F}^F$ cores, and such that $w(F') \leq w^1(A) \leq O(|S|)\,\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$.

Therefore each iteration can be done in polynomial time with probability at least $1-2^{-|E|}$. Lemma 3 ensures that Algorithm 2 terminates after at most $\log |T|$ iterations. Applying a union bound, we get that Algorithm 2 is an $O(|S| \log |T|)$-approximation algorithm for the connectivity augmentation problem, that runs in polynomial time with probability at least $1 - (2^{-|E|} \log |T|)$.

References

1. Byrka, J., Grandoni, F., Rothvoß, T., Sanità, L.: Steiner tree approximation via iterative randomized rounding. Journal of the ACM (JACM) **60**(1), 1–33 (2013)
2. Chan, C., Laekhanukit, B., Wei, H., Zhang, Y.: Polylogarithmic approximation algorithm for k-connected directed steiner tree on quasi-bipartite graphs. In: Byrka, J., Meka, R. (eds.) Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference. LIPIcs, vol. 176, pp. 63:1–63:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPICS.APPROX/RANDOM.2020.63, https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2020.63
3. Charikar, M., Chekuri, C., Cheung, T., Dai, Z., Goel, A., Guha, S., Li, M.: Approximation algorithms for directed Steiner problems. J. Algorithms **33**(1), 73–91 (1999)
4. Cheriyan, J., Laekhanukit, B., Naves, G., Vetta, A.: Approximating rooted steiner networks. ACM Trans. Algorithms **11**(2), 8:1–8:22 (2014). https://doi.org/10.1145/2650183, https://doi.org/10.1145/2650183
5. Dreyfus, S.E., Wagner, R.A.: The steiner problem in graphs. Networks **1**(3), 195–207 (1971). https://doi.org/10.1002/NET.3230010302, https://doi.org/10.1002/net.3230010302
6. Edmonds, J.: Optimum Branchings. Journal of Research of the National Bureau of Standards **71B**, 233–240 (1967)
7. Frank, A.: Increasing the rooted-connectivity of a digraph by one. Math. Program. **84**(3), 565–576 (1999). https://doi.org/10.1007/S101070050040, https://doi.org/10.1007/s101070050040
8. Frank, A.: Rooted k-connections in digraphs. Discret. Appl. Math. **157**(6), 1242–1254 (2009). https://doi.org/10.1016/J.DAM.2008.03.040, https://doi.org/10.1016/j.dam.2008.03.040
9. Frank, A., Tardos, É.: Generalized polymatroids and submodular flows. Math. Program. **42**(1-3), 489–563 (1988). https://doi.org/10.1007/BF01589418, https://doi.org/10.1007/BF01589418
10. Frank, A.: Kernel systems of directed graphs. Acta Sci. Math **41**(1), 63–76 (1979)
11. Garg, N., Konjevod, G., Ravi, R.: A polylogarithmic approximation algorithm for the group steiner tree problem. J. Algorithms **37**(1), 66–84 (2000). https://doi.org/10.1006/JAGM.2000.1096, https://doi.org/10.1006/jagm.2000.1096
12. Goemans, M.X., Goldberg, A.V., Plotkin, S.A., Shmoys, D.B., Tardos, É., Williamson, D.P.: Improved approximation algorithms for network design problems. In: Sleator, D.D. (ed.) Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. 23-25 January 1994, Arlington, Virginia, USA. pp. 223–232. ACM/SIAM (1994), http://dl.acm.org/citation.cfm?id=314464.314497
13. Grandoni, F., Laekhanukit, B.: Surviving in directed graphs: a quasi-polynomial-time polylogarithmic approximation for two-connected directed steiner tree. In: Hatami, H., McKenzie, P., King, V. (eds.) Proceedings of the 49th Annual ACM SIGACT Symposium on Theory

of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017. pp. 420–428. ACM (2017). https://doi.org/10.1145/3055399.3055445, https://doi.org/10.1145/3055399.3055445

14. Gupta, A., Könemann, J.: Approximation algorithms for network design: A survey. Surveys in Operations Research and Management Science **16**(1), 3 – 20 (2011)
15. Hwang, F.K., Richards, D.S.: Steiner tree problems. Networks **22**(1), 55–89 (1992). https://doi.org/10.1002/NET.3230220105, https://doi.org/10.1002/net.3230220105
16. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA. pp. 85–103. The IBM Research Symposia Series, Plenum Press, New York (1972). https://doi.org/10.1007/978-1-4684-2001-2_9, https://doi.org/10.1007/978-1-4684-2001-2_9
17. Karp, R.M., Moreno-Centeno, E.: The implicit hitting set approach to solve combinatorial optimization problems with an application to multigenome alignment. Operations Research **61**(2), 453–468 (2013)
18. Kerivin, H., Mahjoub, A.R.: Design of survivable networks: A survey. Networks **46**(1), 1–21 (2005). https://doi.org/10.1002/NET.20072, https://doi.org/10.1002/net.20072
19. Khuller, S., Vishkin, U.: Biconnectivity approximations and graph carvings. J. ACM **41**(2), 214–235 (1994). https://doi.org/10.1145/174652.174654, https://doi.org/10.1145/174652.174654
20. Klein, P., Ravi, R.: A nearly best-possible approximation algorithm for node-weighted Steiner trees. Journal of Algorithms **19**(1), 104 – 115 (1995)
21. Kortsarz, G., Nutov, Z.: Approximating minimum-cost connectivity problems. In: Gonzalez, T.F. (ed.) Handbook of Approximation Algorithms and Metaheuristics. Chapman and Hall/CRC (2007). https://doi.org/10.1201/9781420010749.CH58, https://doi.org/10.1201/9781420010749.ch58
22. Laekhanukit, B.: Parameters of two-prover-one-round game and the hardness of connectivity problems. In: Chekuri, C. (ed.) Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014. pp. 1626–1643. SIAM (2014). https://doi.org/10.1137/1.9781611973402.118, https://doi.org/10.1137/1.9781611973402.118
23. Laekhanukit, B.: Approximating directed steiner problems via tree embedding. In: Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y., Sangiorgi, D. (eds.) 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy. LIPIcs, vol. 55, pp. 74:1–74:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016). https://doi.org/10.4230/LIPICS.ICALP.2016.74, https://doi.org/10.4230/LIPIcs.ICALP.2016.74
24. Liao, C., Chen, Q., Laekhanukit, B., Zhang, Y.: Almost tight approximation hardness for single-source directed k-edge-connectivity. In: Bojanczyk, M., Merelli, E., Woodruff, D.P. (eds.) 49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France. LIPIcs, vol. 229, pp. 89:1–89:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPICS.ICALP.2022.89, https://doi.org/10.4230/LIPIcs.ICALP.2022.89
25. Nelson, J.: A note on set cover inapproximability independent of universe size. Electronic Colloquium on Computational Complexity (ECCC) **14**(105) (2007)
26. Nutov, Z.: A note on rooted survivable networks. Information Processing Letters **109**(19), 1114 – 1119 (2009)

27. Oliveira, C.A.S., Pardalos, P.M.: A survey of combinatorial optimization problems in multicast routing. Comput. Oper. Res. **32**, 1953–1981 (2005). https://doi.org/10.1016/J.COR.2003.12.007, https://doi.org/10.1016/j.cor.2003.12.007
28. Rothvoß, T.: Directed steiner tree and the lasserre hierarchy. CoRR **abs/1111.5473** (2011), http://arxiv.org/abs/1111.5473
29. Vazirani, V.V.: Approximation algorithms. Springer (2001), http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-65367-7
30. Voß, S.: Steiner tree problems in telecommunications. In: Resende, M.G.C., Pardalos, P.M. (eds.) Handbook of Optimization in Telecommunications, pp. 459–492. Springer (2006). https://doi.org/10.1007/978-0-387-30165-5_18, https://doi.org/10.1007/978-0-387-30165-5_18
31. Zelikovsky, A.: A series of approximation algorithms for the acyclic directed steiner tree problem. Algorithmica **18**(1), 99–110 (1997). https://doi.org/10.1007/BF02523690, https://doi.org/10.1007/BF02523690

A Implicit Hitting Set Randomized Rounding Algorithm

Let $E, w, \mathcal{S}$ be an instance for the implicit hitting set problem. In this setting of implicit hitting set, the size of the input for the problem is polynomial in $|E|$, while $\mathcal{S} \subseteq 2^E$ can be much larger and is thus given implicitly. The fractional relaxation of the problem can be formulated as the following linear program.

$$\begin{array}{llll} min \sum_{e \in E} w_e x_e & s.t. & \sum_{e \in U} x_e \geq 1 & \forall U \in \mathcal{S} \\ & & x_e \geq 0 & \forall e \in E \end{array} \tag{1}$$

A classical result shows that if x^* is an optimal solution to the linear program (1), there is a polynomial time randomized rounding algorithm that returns with probability at least $\frac{1}{2}$ a feasible solution F for the hitting set problem satisfying $w(F) \leq O(\log |\mathcal{S}|) w(x^*)$, where $w(x^*) = \sum_{e \in E} w_e x_e^*$. We can further claim that if there is a polynomial time separation oracle to the linear program (1), then we can efficiently verify whether F is a feasible solution. This result is well known, however for sake of completeness we include the algorithm (presented as Algorithm 3) and formal statement (Lemma 9). For a more detailed analysis see, e.g. [29, Chapter 14].

```
1: let $x^* = \{x_e^*\}_{e \in E}$ be an optimal solution for (1).
2: $F \leftarrow \emptyset$
3: repeat $O(\log |\mathcal{S}|)$ times independently.
4:      For every $e \in E$ independently, add $e$ to $F$ with probability $x_e^*$.
5: if $F$ is a feasible solution to the hitting set problem and $w(F) \leq O(\log |\mathcal{S}|) w(x^*)$
   then
6:    return $F$.
7: else
8:    go to line 2.
```

Algorithm 3: Hitting Set Randomized Rounding

Lemma 9. *Algorithm 3 always returns a feasible solution F of weight at most $O(\log |\mathcal{S}|)$ times the optimal weight. Moreover, if there is a polynomial time separation oracle to the linear program* (1)*, then with probability at least* $1 - 2^{-|E|}$ *Algorithm 3 runs in polynomial time.*

B Proof of Proposition 1

Proof. Let $X, Y \in \mathcal{U}_\ell$, and assume that $X \cap Y \cap T \neq \emptyset$. Therefore $X \cup Y, X \cap Y \in \mathcal{U}$ and furthermore, by submodularity of the cut function,

$$2\ell = |\varrho_{E_\ell}(X)| + |\varrho_{E_\ell}(Y)| \geq |\varrho_{E_\ell}(X \cup Y)| + |\varrho_{E_\ell}(X \cap Y)| \ .$$

By the definition of E_ℓ, $|\varrho_{E_\ell}(X \cup Y)| = |\varrho_{E_\ell}(X \cap Y)| = \ell$, and therefore $X \cap Y, X \cup Y \in \mathcal{U}_\ell$.

C Proofs for Section 3.1

In this section we complete the technical proofs of Section 3.1.

C.1 Proof of Lemma 2

Proof. Assume $C \cap X \cap T \neq \emptyset$, then by the previous claim, $C \cap X \in \mathcal{F}^F$. Since $C \in \mathcal{F}^F$ is inclusion-minimal, $C \cap X = C$ and thus $C \subseteq X$.

C.2 Proof of Lemma 3

Proof. Let $C \in \mathcal{M}(\mathcal{F}^{F \cup F'})$, then $C \in \mathcal{F}$ is not covered by $F \cup F'$, i.e. $\varrho_{F \cup F'}(C) = \emptyset$. It follows that $\varrho_F(C) = \emptyset$, and therefore $C \in \mathcal{F}^F$. In addition, $\varrho_{F'}(C) = \emptyset$ and thus C is not covered by F'. Since F' covers all $\mathcal{F}^F$-cores, C is not an $\mathcal{F}^F$-core. Hence, there are $X, Y \in \mathcal{M}(\mathcal{F}^F)$ such that $X, Y \subseteq C$. Denote $\mathcal{X}_C := \{X \in \mathcal{M}(\mathcal{F}^F) : X \subseteq C\}$. Then $\mathcal{X}_C \subseteq \mathcal{M}(\mathcal{F}^F)$, and we have shown that $|\mathcal{X}_C| \geq 2$. Since $\mathcal{F}^{F \cup F'}$ is also a T-intersecting family, inclusion-minimal elements of $\mathcal{F}^{F \cup F'}$ are terminal-disjoint. Therefore C is the unique element of $\mathcal{M}(\mathcal{F}^{F \cup F'})$ containing X, Y. It follows that $\{\mathcal{X}_C\}_{C \in \mathcal{M}(\mathcal{F}^{F \cup F'})}$ is a partition of $\mathcal{M}(\mathcal{F}^F)$, in which every set contains at least two elements. Therefore $|\mathcal{M}(\mathcal{F}^{F \cup F'})| \leq \frac{1}{2}|\mathcal{M}(\mathcal{F}^F)|$.

C.3 G^F can be constructed in polynomial time

In order to prove that G^F can be constructed efficiently, we need to show that we can efficiently decide for every $x \in V$ whether $x \in \bigcup_{C \in \mathcal{M}(\mathcal{F}^F)} C$.

Claim. Let $t \in T$. If there exists a tight set $X \in \mathcal{F}^F$ such that $t \in X$, then there exists a unique inclusion-minimal such set.

Proof. Denote $R_t = \{X \in \mathcal{F}^F : t \in X\}$. Since $t \in T$, and since $\mathcal{F}^F$ is T-intersecting, we get that $\bigcap\limits_{X \in R_t} X \in \mathcal{F}^F$. Clearly it is an inclusion-minimal such set, and we similarly show it is unique.

For every $t \in T$, denote by C_t the unique inclusion-minimal tight set containing t. The following claim is straightforward, and it characterizes these minimal sets in a manner that allows to recover them efficiently given G, E_ℓ, F.

Claim. For every $t \in T$, there exists a tight set $X \in \mathcal{F}^F$ containing t if and only if the minimum rt-cut in $(V, E_\ell \cup F)$ contains ℓ edges. Moreover, C_t is an inclusion minimal rt-cut of capacity ℓ and can therefore be computed in polynomial time.

Intuitively, one may think that for every $t \in T$, C_t is an inclusion-minimal element of $\mathcal{F}^F$. However, it turns out that this needs not be the case. Therefore in general we cannot argue that $\{C_t\}_{t \in T} \subseteq \mathcal{M}(\mathcal{F}^F)$. The following claim shows that the converse containment does hold.

Proposition 2. *$\mathcal{M}(\mathcal{F}^F) \subseteq \{C_t\}_{t\in T}$. Furthermore, for every $t \in T$, $C_t \in \mathcal{M}(\mathcal{F})$ if and only if there exists a strict core containing t.*

Proof. Let $C \in \mathcal{M}(\mathcal{F}^F)$, and let $t \in C \cap T$. Then C is an inclusion-minimal F-tight set containing t. Therefore $C = C_t \in \{C_s\}_{s\in T}$. Next, fix some $t \in T$. Clearly, if $C_t \in \mathcal{M}(\mathcal{F})$, then C_t is a strict core containing t. Assume that there exists a strict core X containing t. Then there exists $C \in \mathcal{M}(\mathcal{F}^F)$ such that $C \subseteq X$ and $X \cap T = C \cap T$. Since $t \in C_t \cap X \cap T = C_t \cap C \cap T$, and by minimality of both C and C_t, and since $C_t \cap C$ is tight, it follows that $C_t \cap C = C_t = C \in \mathcal{M}(\mathcal{F}^F)$.

We are now ready to prove the main claim of this section.

Proof. We first show that E^0 (and thus w^0) can be constructed in polynomial time given G, E_ℓ, F. Following the characterization of $\{C_t\}_{t\in T}$ and $\mathcal{M}(\mathcal{F}^F)$ in Claim C.3 and Proposition 2, we can construct the family $\mathcal{M}(\mathcal{F}^F)$ in polynomial time. Let $u, v \in V$, and consider the following algorithm for constructing E^0.

```
for all $C \in \mathcal{M}(\mathcal{F}^F)$ do
    find in $(V, E_\ell \cup F)$ a minimum cut $U_C$ that separates $C \cup \{v\}$ and $\{r, u\}$.
    if $\min_{C\in\mathcal{M}(\mathcal{F}^F)} |\varrho_{E_\ell \cup F}(U_C)| \geq \ell + 1$ then
        add $uv$ to $E^0$.
```

Note first that the algorithm runs in polynomial time. We will show that the algorithm adds uv to E^0 if and only if for every $X \in \mathcal{F}^F$, if $v \in X$ then $u \in X$. Assume first that there exists $X \in \mathcal{F}^F$ such that $v \in X$ and $u \notin X$. Then there exists some $C \in \mathcal{M}(\mathcal{F}^F)$ such that $\{v\} \cup C \subseteq X \subseteq V \setminus \{r, u\}$. Therefore X is a cut separating $C \cup \{v\}$, and since $X \in \mathcal{F}^F$, then the capacity of X is ℓ. Therefore $\varrho_{E_\ell \cup F}(U_C) \leq \ell$ and uv is not added to E^0. Otherwise, for every $X \in \mathcal{F}^F$, if $v \in X$ then $u \in X$. Let $C \in \mathcal{M}(\mathcal{F}^F)$, and let U_C be a minimum cut that separates $C \cup \{v\}$ and $\{r, u\}$ in $(V, E_\ell \cup F)$. Then $C \subseteq U_C \subseteq V \setminus \{r\}$, and therefore $U_C \in \mathcal{U}$. Since $v \in U_C$ and $u \notin U_C$, then $U_C \notin \mathcal{F}^F$. Therefore $\varrho_{E_\ell \cup F}(U_C) \geq \ell + 1$. Hence uv is added to E^0 by the algorithm.

Given G^0, the metric completion G^1 can be constructed in polynomial time. Finally, for every $u \in V$ and $s \in T$, by Proposition 2 we can decide in polynomial time whether $u \notin \bigcup_{C\in\mathcal{M}(\mathcal{F}^F)} C$ and whether $s \in \bigcup_{C\in\mathcal{M}(\mathcal{F}^F)} C$. In a similar manner to constructing E^0, we can decide efficiently whether to remove all edges leaving u. Therefore the set $E(G^F)$ can be constructed in polynomial time.

C.4 Proof of Lemma 8

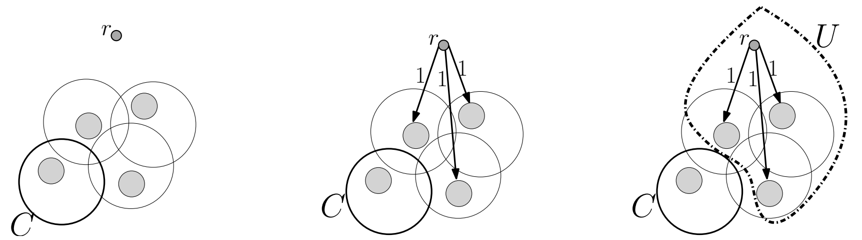
Consider the instance $E(G^F), w, \mathcal{S}^F$ for the implicit hitting set problem, where

$$\mathcal{S}^F = \{\varrho_{E(G^F)}(X) : X \text{ is a strict } \mathcal{F}^F\text{-core}\} ,$$

and the corresponding fractional relaxation.

$$\begin{array}{llll} min \sum_{e \in E(G^F)} w^1_e x_e & s.t. & \sum_{e \in \varrho_{E(G^F)}(X)} x_e \geq 1 & \text{for all strict } \mathcal{F}^F\text{-cores } X \\ & & x_e \geq 0 & \forall e \in E(G^F) \end{array} \tag{2}$$

We present a polynomial time separation oracle for the linear program (2). That is, we show an algorithm which, given a vector $x \in \mathbb{R}_+^{E(G^F)}$ either reports that x is feasible for (2) or returns a constraint violated by x. For every $t \in T$, the algorithm verifies that x satisfies the constraints regarding strict cores containing t. Given some $t \in T$, we first check whether there is a strict core containing t. If there is no such strict core, we are done. Otherwise, we find the (unique) element $C \in \mathcal{M}(\mathcal{F}^F)$ containing t, and construct an edge-capacitated auxiliary graph $H_t = (V, E', \{x^t_e\}_{e \in E'})$, where $E' = E(G^F) \cup E_\ell \cup F$ (note that E' may contain parallel edges), and x^t is defined as follows. For every $s \in T \setminus C$ we set x^t_{rs} to 1, as demonstrated in Figure 1. In addition, we define $x^t_e = 1$ for all $e \in E_\ell \cup F$. For all other edges e, we set $x^t_e = x_e$. We now find a minimum rt-cut in H_t. The key observation is that if the capacity of the cut is strictly less than $\ell + 1$, then $V \setminus U$ is a strict $\mathcal{F}^F$-core that is violated by x (see Figure 1c). The detailed algorithm is given as Algorithm 4.



(a) Given $t \in T$, the highlighted area represents the unique $C \in \mathcal{M}(\mathcal{F}^F)$ such that $t \in C$. The white circles represent $\mathcal{M}(\mathcal{F}^F) \setminus \{C\}$, while their intersection with T is dark.

(b) We add 1-capacity edges from r to s for all $s \in T \setminus C$.

(c) An rt-cut U of capacity strictly less than 1 must then contain $T \setminus C$. Therefore the set $V \setminus U$ is a strict $\mathcal{F}^F$-core.

Fig. 1: Constructing an Auxiliary Graph H_t

We first show that the Algorithm 4 runs in time polynomial in the size of G. Fix some $t \in T$. We can efficiently find C_t, and by the characterization in Proposition 2, we can efficiently check if $C_t \in \mathcal{M}(\mathcal{F}^F)$. Since adjusting the graph and finding a minimum cut can be done efficiently, we get that Algorithm 4

```
Input: $x = \{x_e\}_{e \in E(G^F)}$
Output: either report “$x$ is feasible” or find a violated constraint of (2).
let $E' = E(G^F) \cup E_\ell \cup F$ be a multiset of edges.
for all $t \in T$ do
    if $C_t \in \mathcal{M}(\mathcal{F}^F)$ then
        for all $e \in E'$ do
            if $e \in F \cup E_\ell$ or there exists $s \in T \setminus C_t$ such that $e = rs$ then
                let $x^t_e \leftarrow 1$.
            else
                let $x^t_e \leftarrow x_e$.
        find a minimum $rt$-cut $U$ in $H_t = (V, E')$ with capacities $\{x'_e\}_{e \in E'}$.
        if the capacity of the cut is less than $\ell + 1$ then
            return $V \setminus U$.
return $x$ is feasible.
```

Algorithm 4: Separation Oracle for (2)

runs in polynomial time. The following two claims prove the correctness of the algorithm.

Claim. If x is feasible for (2), then Algorithm 4 verifies it.

Proof. Assume x is feasible for (2), and let $t \in T$ be such that $C_t \in \mathcal{M}(\mathcal{F}^F)$. Let U be an rt-cut in H_t, and denote $X = V \setminus U$. Then $t \in X$ and $r \notin X$, and therefore $X \in \mathcal{F}$. If there is an edge $e \in F$ that covers X, then $e \in \varrho_{E'}(X)$, and the capacity of the cut is $\sum_{e' \in \varrho_{E'}(X)} x^t_{e'} \geq x^t_e + \sum_{e' \in \varrho_{E_\ell}(X)} x^t_{e'} = \ell + 1$. If there is $s \in T \setminus C_t$ such that $s \in X$, then similarly $\sum_{e' \in \varrho_{E'}(X)} x^t_{e'} \geq \ell + 1$. Otherwise, $X \in \mathcal{F}^F$, and moreover, $X \cap T = C_t \cap T$. Therefore X is a strict $\mathcal{F}^F$-core, and since x is feasible for (2),

$$\sum_{e' \in \varrho_{E'}(X)} x^t_{e'} \geq \ell + \sum_{e' \in \varrho_{E(G^F)}(X)} x^t_{e'} = \ell + \sum_{e' \in \varrho_{E(G^F)}(X)} x_{e'} \geq \ell + 1 \ .$$

Claim. If x is infeasible for (2), then Algorithm 4 returns a violated strict $\mathcal{F}^F$-core.

Proof. Assume that x is infeasible. Then there exists $X \in \mathcal{F}^F$ such that X is a strict $\mathcal{F}^F$-core and such that

$$\sum_{e \in \varrho_{E(G^F)}(X)} x_e < \ell + 1 \ .$$

Let $t \in X \cap T$, then $(V \setminus X, X)$ is an rt-cut of capacity < 1 in $(V, E(G^F), \{x_e\}_{e \in E(G^F)})$. Note that since X is a strict core, no edge in F enters X. Therefore $(V \setminus X, X)$ is an rt-cut of capacity $< \ell + 1$ in H_t. We conclude that the capacity of the minimum rt-cut U found by the algorithm is strictly less than $\ell + 1$. It remains to show that $Y = V \setminus U$ is a strict $\mathcal{F}^F$-core. Clearly $t \in Y \cap T$ and $r \notin Y$. Let $e \in F$, then $x_e^t = 1$. Since

$$0 \leq |\varrho_F(Y)| = \sum_{e \in \varrho_F(Y)} x_e^t \leq \sum_{e \in \varrho_{E'}(Y)} x_e^t - \ell < 1 \ ,$$

then $Y \in \mathcal{F}^F$. Consider $s \in Y \cap T$. If $s \notin C$, then $rs \in \varrho_{E'}(Y)$ and thus $\sum_{e \in \varrho_{E'}(Y)} x_e^t \geq \ell + x_{rs}^t = \ell + 1$. Therefore C is the unique element of $\mathcal{M}(\mathcal{F}^F)$ contained in Y, and $Y \cap T = C \cap T$.

Corollary 1. *Algorithm 4 is a polynomial time separation oracle for the linear program* (2)

Proof (Proof of Lemma 8). From Corollary 1 and Lemma 9 we deduce that the randomized rounding algorithm for the hitting set problem (Algorithm 3) outputs a set $A \subseteq E(G^F)$ such that $w^1(A) \leq O(\log |\mathcal{S}^F|)\,\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$. Following Lemma 4, $|\mathcal{S}^F| \leq 2^{|S|}|T|$, and therefore $w^1(A) \leq O(|S|)\,\mathrm{opt}_{\mathrm{aug}}(G, E_\ell)$. Moreover, Lemma 9 guarantees that with probability at least $1 - 2^{-|E|}$ the algorithm runs in polynomial time.