Surviving in Directed Graphs: A Polylogarithmic Approximation for Two-Connected Directed Steiner Tree

Fabrizio Grandoni* Bundit Laekhanukit†

November 8, 2016

Abstract

Real-word networks are often prone to failures. A reliable network needs to cope with this situation and must provide a backup communication channel. This motivates the study of *survivable network design*, which has been a focus of research for a few decades. To date, survivable network design problems on undirected graphs are well-understood. For example, there is a 2 approximation in the case of edge failures [Jain, FOCS'98/Combinatorica'01]. The problems on *directed graphs*, in contrast, have seen very little progress. Most techniques for the undirected case like primal-dual and iterative rounding methods do not seem to extend to the directed case. Almost no non-trivial approximation algorithm is known even for a simple case where we wish to design a network that tolerates a single failure.

In this paper, we study a survivable network design problem on directed graphs, 2-Connected Directed Steiner Tree (2-DST): given an n-vertex weighted directed graph, a root r, and a set of h terminals S, find a min-cost subgraph H that has two edge/vertex disjoint paths from r to any $t \in S$. 2-DST is a natural generalization of the classical Directed Steiner Tree problem (DST), where we have an additional requirement that the network must tolerate one failure. No non-trivial approximation is known for 2-DST. This was left as an open problem by Feldman et al., [SODA'09; JCSS] and has then been studied by Cheriyan et al. [SODA'12; TALG] and Laekhanukit [SODA'14]. However, no positive result was known except for the special case of a D-shallow instance [Laekhanukit, ICALP'16].

We present an $O(D^3 \log D \cdot h^{2/D} \cdot \log n)$ approximation algorithm for 2-DST that runs in time $O(n^{O(D)})$, for any $D \in [\log_2 h]$. This implies a polynomial-time $O(h^\varepsilon \log n)$ approximation for any constant $\varepsilon > 0$, and a poly-logarithmic approximation running in quasi-polynomial time. We remark that this is essentially the best-known even for the classical DST, and the latter problem is $O(\log^{2-\varepsilon} n)$-hard to approximate [Halperin and Krauthgamer, STOC'03]. As a by product, we obtain an algorithm with the same approximation guarantee for the 2-Connected Directed Steiner Subgraph problem, where the goal is to find a min-cost subgraph such that every pair of terminals are 2-edge/vertex connected.

Our approximation algorithm is based on a careful combination of several techniques. In more detail, we decompose an optimal solution into two (possibly not edge disjoint) *divergent trees* that induces two edge disjoint paths from the root to any given terminal. These divergent trees are then embedded into a shallow tree by means of Zelikovsky's height reduction theorem. On the latter tree we solve a 2-Connected Group Steiner Tree problem and then map back this solution to the original graph. Crucially, our tree embedding is achieved via a probabilistic mapping guided by an LP: This is the main technical novelty of our approach, and might be useful for future work.

*IDSIA, University of Lugano, Switzerland, fabrizio@idsia.ch. Partially supported by the ERC Starting Grant NEWNET 279352 and the SNSF Grant APPROXNET 200021_159697/1.

†Weizmann Institute of Science, Israel, bundit.laekhanukit@weizmann.ac.il. Partially supported by the ISF (grant No. 621/12) and by the I-CORE Program (grant No. 4/11).

1 Introduction

Real-world networks are often prone to link or node failures. A reliable network needs to cope with this situation and must provide a backup communication channel. In mathematical terms, ones wish to design a network that provides a pre-specified number of edge/vertex disjoint paths between given pairs of *terminals*. This motivates the study of *survivable network design*, which has been a focus of research for a few decades [45, 18, 23, 28].

To date, the survivable network design problems on undirected graphs are well-understood, and many powerful techniques have been developed to solve this class of problems. For example, in the edge failure case, there is a 2-approximation algorithm by Jain [28] for the most general version of the problem, *Generalized Steiner Network*. In contrast, there has been very slow progress on survivable network design in *directed graphs*. Most of the standard techniques like primal-dual and iterative rounding methods do not seem to extend to the directed case. Positive results are known only for very restricted cases (see, e.g., [12, 37, 20, 36]). In fact, there are almost no positive results for survivable network design on directed graphs in the present of Steiner vertices.

In this paper, we focus on arguably one of the simplest survivable network design problems in directed graphs, namely, 2-Connected Directed Steiner Tree (2-DST):

Definition 1. *In the* 2*-connected Directed Steiner Tree problem (*2*-DST), we are given an* n*-vertex directed graph* $G = (V, E)$ *with edge-costs* $\{c_e\}_{e \in E}$*, a root vertex* r *and a set of* h *terminals* $S \subseteq V - \{r\}$*. The goal is to find a min-cost subgraph* H *that has at least* 2 *edge disjoint paths from* r *to each* $t \in S$*.*

Intuitively, the goal of 2-DST is to design a network that can function in the event of one edge failure (thus, it must provide a backup path). 2-DST is a natural generalization of the classical Directed Steiner Tree problems (DST), where only one r, t-path for each $t \in S$ is required to exist in H. Feldman et al. [16] left approximating 2-DST as an open problem (see also the earlier work in [12]), and the problem has later been studied in the work of Cheriyan et al. [8] and Laekhanukit [34, 36]. However, there was no known non-trivial approximation algorithm for 2-DST except for the special case of D-shallow instances (where the length of any root-to-terminal path in the optimal solution is at most D[1]) [36].

Here we define 2-DST in terms of edge-connectivity. The vertex-connectivity version is defined analogously, but we are asked for vertex-disjoint instead of edge-disjoint paths. The two variants share the same approximability in directed graphs. There is a simple reduction that reduces the vertex-connectivity version to edge-connectivity version[2] and vice versa. We will therefore focus only on the edge connectivity case.

1.1 Our Results and Techniques

The main contribution of this paper is a non-trivial approximation algorithm for 2-DST.

Theorem 2. *For any* $D \in [\log_2 h]$*, there exists a randomized* $O(D^3 \log D \cdot h^{2/D} \cdot \log n)$ *approximation algorithm for* 2*-DST that runs in* $n^{O(D)}$ *time.*

[1] A D-shallow instance is an instance that has an optimal solution H such that, for every terminal t, H has k edge-disjoint r, t-paths in which each path has length at most D (i.e., all the k paths are short). This imitates the notion of the height of a tree, but it allows H to contain a directed cycle.

[2] In more detail, split each vertex v into v^{in} and v^{out}, add a zero-cost edge $v^{in}v^{out}$, and then re-wire each edge entering and leaving v to v^{in} and v^{out}, respectively. A source-sink pairs (s, t) is then replaced by the pair (s^{out}, t^{in}). The number of pairs does not change, and the number of vertices grows by a factor 2.

In particular, Theorem 2 implies a polynomial-time $O(h^{\varepsilon} \log n)$ approximation for any constant $\varepsilon > 0$, and a quasi-polynomial-time $O(\log n \log^3 h \log \log h)$ approximation algorithm. We remark that, up to poly-logarithmic factors, this is the best known even for the simpler case of DST [6].

Approximation algorithms for 2-DST can be used to approximate with the same asymptotic approximation factor the more general problem, namely 2-*DSS*, described in [11, 35, 41] (see Appendix A for more details).

Definition 3. *In the* 2*-Connected Directed Steiner Subgraph problem (2-DSS), we are given a directed graph* $G = (V, E)$ *with edge-costs* $\{c_e\}_{e \in E}$ *and a set of terminals* $S \subseteq V$*. The goal is to find a min-cost subgraph* H *of* G *such that, for every pair of vertices* $s, t \in S$*,* H *has* 2 *edge-disjoint paths from* s *to* t*.*

As a corollary of Theorem 2, we obtain the following result.

Corollary 4. *For any* $D \in [\log_2 h]$*, there exists a randomized* $O(D^3 \log D \cdot h^{2/D} \cdot \log n)$ *approximation algorithm for 2-DSS that runs in* $n^{O(D)}$ *time.*

Our approach is rather sophisticated, and involves several logical steps. The starting point is the following decomposition theorem[3].

Theorem 5 (Divergent Steiner Trees Theorem [22, 33])**.** *Let* H *be a feasible solution to a* 2*-DST instance with a root* r *and terminals* S*. Then* H *can be decomposed into two (possibly overlapping) arborescences (*divergent Steiner trees*)* T_1 *and* T_2 *rooted at* r *and spanning* S *such that, for every terminal* $t \in S$*, the unique* r*-*t *paths* P_1 *in* T_1 *and* P_2 *in* T_2 *are edge disjoint.*

Intuitively, T_1 and T_2 are two solutions to the DST problem on the same instance with the extra property of being edge disjoint "from the point of view" of a single terminal. We remark that this is the only part of our approach that does not directly generalize to connectivity $k \geq 3$ because the decomposition does not exist for $k \geq 3$ [27, 3]. (See the discussion in Section 5.)

The second main tool from the literature that we wish to exploit is the Zelikovsky's height-reduction theorem [26, 46] that is used in approximating DST.

Theorem 6 (Height Reduction Theorem [26, 46])**.** *Consider an edge weighted arborescence* T *rooted at* r *and spanning* S*. Then, for any* $D \in [\log_2 |S|]$*, in the metric completion of* T*, there exists an arborescence* T^D *of depth at most* D *rooted at* r *and spanning* S *together with a mapping* ψ *that maps each vertex of* T^D *to a vertex of* T *and a mapping* ϕ *that maps each edge* $\hat{e} = \hat{u}\hat{v} \in E(T^D)$ *to a* $\psi(\hat{u}), \psi(\hat{v})$ *path* $\phi(\hat{e})$ *in* T *so that the following* bounded congestion *property holds:*

$$|\hat{e} \in E(T^D) : e \in \phi(\hat{e}))\}| \leq \beta' = O(D \cdot |S|^{1/D}) \quad \forall e \in E(T).$$

In particular, the cost of T^D *is at most* β' *times the cost of* T*.*

We remark that the Height Reduction Theorem was originally stated in terms of cost (which is implied by our version). Here we extract the bounded congestion property that is implicit in the proof.

The main difficulty that we have to face is how to apply these two tools. In DST approximation, Theorem 6 is typically applied by considering the metric closure of the input graph. This is not applicable to our case since the metric closure might destroy the connectivity properties of the input graph. Moreover, we cannot directly apply the theorem to the divergent Steiner trees because they are a decomposition of *an optimal solution* that we wish to compute.

[3]There also exists a vertex-connectivity analogue of this theorem, but we omit it here since it is not necessary for our goals.

We solve these issues by defining an ILP that mimics the decomposition of the optimal solution into divergent Steiner trees T_1 and T_2 (as in Theorem 5) and the following application of Theorem 6 to these trees to obtain D-shallow trees T_1^D and T_2^D. In more detail, we define a D-shallow tree that incorporates (twice) all the possible paths of length D starting from the root (analogously to [36]). This shallow tree implicitly includes T_1^D and T_2^D. We encode the mapping of each edge of $T_1^D \cup T_2^D$ into the associated paths in $T_1 \cup T_2$ using flow constraints. We also add constraints that encode the *bounded congestion* property from Theorem 6 (crucial to bound the cost of the approximate solution) and the *divergency* property from Theorem 5 (crucial to achieve a feasible solution).

Rounding a fractional solution to the linear relaxation is a non-trivial task. We observe that each terminal $t \in S$ is associated with a subset of vertices $\hat{S}_t$ in the shallow tree, and the edges of $T_1^D \cup T_2^D$ must contain two edge disjoint paths from the root $\hat{r}$ to $\hat{S}_t$. In other words, the latter edges induce a feasible solution to a tree instance of 2-Edge Connected Group Steiner Tree (2-GST) with root $\hat{r}$ and groups $\{\hat{S}_t\}_{t\in S}$ (more details in related work). This allows us to add the standard LP constraints for 2-GST on *a tree* to our linear relaxation, and use the GKR rounding algorithm by Garg et al. [21] to round the corresponding variables to an integral 2-GST solution in the shallow tree.

The last obstacle that we need to face is that we need to map back each chosen edge $\hat{e}$ of the shallow tree to a path $\phi(\hat{e})$ of the original graph. The LP solution provides a *fractional mapping* in the form of a flow. We interpret this flow as a distribution over paths and sample one path $\phi(\hat{e})$ according to this distribution. In order to show that the solution is feasible (with large enough probability), we exploit an argument similar in spirit to the one used by Chalermsook et al. [5] in the framework of k-Edge Connected Group Steiner Tree (k-GST) approximation. However, our probabilistic mapping makes the analysis slightly more involving. Shortly, we argue that for any given edge e of the original graph, GKR rounding has sufficiently large probability to select paths using only edges $\hat{e}$ of the shallow tree whose associated probabilistic mapping has low chance to use the edge e. The claim then follows by a cut argument as in [5].

1.2 Related Work

In the Directed Steiner Tree problem (DST), we are given an n-vertex directed edge weighted graph, a root r and a collection of h terminal vertices S. The goal is to find a min-cost arborescence rooted at r and spanning S. DST is one of the most fundamental network design problems in directed graphs. DST admits, for any positive integer D, an $O(Dh^{1/D}\log^2 h)$ approximation running in time $n^{O(D)}$ [6, 46]. In particular, this implies a polynomial-time $O(h^{\varepsilon})$ approximation for any constant $\varepsilon > 0$, and an $O(\log^3 h)$ approximation running in quasi-polynomial time.

k-DST and k-DSS are the natural generalization of 2-DST and 2-DSS, respectively, with connectivity k. These problems have been a subject of study since early 90's [12] and have been subsequently studied in [8, 34, 36]. Cheriyan et al. [8] showed that k-DST is at least as hard as the *Directed Steiner Forest* problem and the *Label-Cover* problem. Thus, k-DST admits no $2^{\log^{1-\varepsilon} n}$-approximation algorithm, for any $\varepsilon > 0$, unless $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{\mathrm{polylog}(n)})$. For small k, they showed that k-DST admits no k^{σ}-approximation algorithm for some fixed $\sigma > 0$ unless $\mathrm{P} = \mathrm{NP}$. If k is large enough, then k-DST is NP-hard even when we have only two terminals, and they further proved that k-DST when h and k are constants is polynomial-time solvable in directed acyclic graphs. However, if the input graph contains a cycle, the complexity status of k-DST is not clear even for $k = h = 2$. Laekhanukit refined the hardness result of k-DST in [34] and showed that k-DST admits no $k^{1/2-\varepsilon}$-approximation algorithm, for any constant $\varepsilon > 0$, unless $\mathrm{NP} = \mathrm{ZPP}$. In a subsequent work, Laekhanukit [36] presented an LP-based $O(k^{D-1}D\log n)$-approximation algorithm for *D-shallow* instances of k-DST and k-DSS running in time $n^{O(D)}$. It seems that his approach cannot be generalized to arbitrary instances (although we will exploit part of his ideas).

A well-studied special case of DST is the *Group Steiner Tree* problem (GST). Here we are given an undirected weighted graph, a root vertex r, and a collection of h groups $S_i \subseteq V$. The goal is to compute the cheapest tree that spans r and at least one vertex from each group S_i. The best-known polynomial-time approximation factor for GST is $O(\log^2 h \log n)$ due to Garg et al. [21]. Their algorithm uses *probabilistic distance-based tree embeddings* [2, 15] as a subroutine. Chekuri and Pal [7] presented an $O(\log^2 h)$ approximation that runs in quasi-polynomial time. On the negative side, Halperin and Krauthgamer [25] showed that GST admits no $\log^{2-\varepsilon} n$-approximation algorithm, for any constant $\varepsilon > 0$, unless $\mathrm{NP} \subseteq \mathrm{ZPTIME}(2^{\mathrm{polylog}(n)})$. This implies the same hardness for DST, hence for 2-DST and 2-DSS.

The high-connectivity version of GST, namely, the *k-Edge Connected Group Steiner Tree problem* (k-GST), was studied in [5, 24, 29]. Here the goal is to find a min-cost subgraph that contains k edge-disjoint paths between the root and each group. For $k = 2$, the best approximation ratio is $\tilde{O}(\log^3 n \log h)$ due to the work of Gupta et al. [24]. If the size of any group is bounded by α, then there is an $O(\alpha \log^2 n)$-approximation algorithm by Khandekar et al. [29]. For $k \geq 3$, there is no known non-trivial approximation algorithm for k-GST. Chalermsook et al. [5] presented an LP-rounding bicriteria approximation algorithm that returns a subgraph with cost $O(\log^2 n \log h)$ times the optimum while guarantees a connectivity of at least $\Omega(k/\log n)$. Their algorithm uses the *probabilistic cut-based tree embeddings* by Räcke [43] as a subroutine (as opposed to distance-based ones in [24]). We will exploit part of their ideas in our rounding algorithm (although a probabilistic tree embedding for directed graphs is not available for us). Chalermsook et al. also showed that k-GST is hard to approximate to within a factor of k^σ, for some fixed constant $\sigma > 0$, and if k is large enough, then the problem is at least as hard as the Label-Cover problem, meaning that k-GST admits no $2^{\log^{1-\varepsilon} n}$-approximation algorithm, for any constant $\varepsilon > 0$, unless $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{\mathrm{polylog}(n)})$.

As already mentioned, survivable network design is well studied in undirected (weighted) graphs. First, consider the edge connectivity version. The earliest work is initiated in early 80's by Frederickson and JáJá [18], where the authors studied the *2-Edge Connected Subgraph* problem in both directed and undirected graphs. In the most general form of the problem, also known as the *Steiner Network* problem, we are given non-negative integer requirements $k_{u,v}$ for all pairs of vertices u, v, and the goal is to find a min-cost subgraph that has $k_{u,v}$ edge-disjoint paths between u and v. Jain [28] devised a 2-approximation algorithm for this problem. We remark that 2 is the best known approximation factor even for $k_{u,v} \in \{0, 1\}$ [1], which is known as the *Steiner forest* problem. The classical *Steiner tree* problem is a special case of Steiner forest where all pairs share one vertex. Here the best known approximation factor is 1.39 due to the work of Byrka et al. [4].

Concerning vertex connectivity, two of the most well-studied problems are the *k-Vertex Connected Steiner Tree* (k-ST) and *k-Vertex Connected Steiner Subgraph* (k-SS) problems, i.e., the undirected versions of k-DST and k-DSS, respectively. There are 2-approximation algorithms for 2-ST and 2-SS by Fleischer et al. [17] using the iterative rounding method. For $k \geq 3$, Nutov devised an $O(k \log k)$-approximation algorithm for k-ST in [40] and an $O(\min\{|S|^2, k \log^2 k\})$-approximation algorithm for k-SS in [41] (also, see [35]). A special case of k-SS with metric-costs is studied by Cheriyan and Vetta in [11] who gave an $O(1)$-approximation algorithm for the problem. The most extensively studied special case of k-SS is when all vertices are terminals, namely the *k-Vertex Connected Spanning Subgraph* problem, which has been studied, e.g., in [10, 31, 14, 42, 9]. The current best approximation guarantees are $O(\log(n/(n-k)) \log k)$ [42], and 6 for the case $n \leq 2k^3$ [9, 19]. More references can be found in [32, 38, 39].

Notation. We use standard graph terminologies. For any graph G, we denote vertex and edge sets of G by $V(G)$ and $E(G)$, respectively. For any subset of vertices $S \subseteq V(G)$ (or a single vertex $S = v$), we denote the set of edges of G entering S by $\delta^{in}_G(S)$ and denote the set of edges leaving S by $\delta^{out}_G(S)$.

2 Embedding into a Shallow Tree

Our LP-relaxation is defined based on the existence an embedding of an optimal 2-DST solution H in the original graph into an auxiliary D-shallow tree $\hat{H}$ (i.e., a tree of depth at most D), where $D > 0$ is an integer given as parameter. Our embedding is obtained by applying the *Height Reduction* to *Divergent Steiner Trees*.

We start by decomposing $\hat{H}$ into two divergent Steiner trees T_1 and T_2 using the Divergent Steiner Tree Theorem (Theorem 5). Then we apply the Height Reduction Theorem (Theorem 6) to each such T_i, hence getting a D-shallow tree T_i^D in the metric closure of T_i together with mappings ψ_i and ϕ_i. The final step is to unify the roots of T_1^D and T_2^D, hence getting a tree $\hat{H}$ rooted at $\hat{r}$. We also merge the two mappings in a natural way, thus getting $\psi : V(\hat{H}) \to V(H)$ and $\phi : E(\hat{H}) \to 2^{E(H)}$. Let $\psi^{-1}(v)$ be the set of vertices $\hat{v} \in V(\hat{H})$ with $\psi(\hat{v}) = v$. Note also that each simple $\hat{u}, \hat{v}$-path $\hat{P}$ in $\hat{H}$ defines a $\psi(\hat{u}), \psi(\hat{v})$ path $P = \phi(\hat{P})$ in H.

By construction, it is not hard to see that $(\hat{H}, \psi, \phi)$ has the following properties:

1. (*divergency*) for any terminal $t \in S$, there exist two vertices $\hat{t}_1, \hat{t}_2 \in \psi^{-1}(t)$ such that the following holds. Let $\hat{P}_i$ be the $\hat{r}$-$\hat{t}_i$ path in $\hat{H}$ for $i = 1, 2$. Then $\phi(\hat{P}_1)$ and $\phi(\hat{P}_2)$ are two edge-disjoint r-t paths in H (and consequently also in $\hat{H}$).
2. (*bounded congestion*) For any edge $e \in E(H)$, $|\hat{e} \in E(\hat{H}) : e \in \phi(\hat{e})| \leq \beta := 2\beta' = O(D|S|^{1/D})$.

Note that we do not know an optimal solution, and consequently the two trees T_1, T_2 that are needed to define the above embedding. In the next section, we define an LP relaxation that, in some fractional sense, achieves this goal.

3 An LP-relaxation for 2-DST

In this section, we present an ILP formulation of 2-DST, and the corresponding LP relaxation.

The first step in the definition of our ILP is to build a proper D-shallow tree $\hat{T} = (\hat{V}, \hat{E})$ that contains the tree $\hat{H}$ (defined in the previous section) as a subgraph. To this end, we list *twice* all the possible sequences of at most $D + 1$ distinct vertices of G starting with the root r. The *prefix tree* of these sequences (rooted at $\hat{r} = r$) is our tree $\hat{T}$. That is, each vertex $\hat{v}$ of $\hat{T}$ is associated with a vertex v in the input graph G, and each rooted-path in $\hat{T}$ corresponds to each sequence we listed. It is not hard to see that $\hat{H}$ can be mapped to a subtree of $\hat{T}$. Let $\psi : \hat{V} \to V$ be the corresponding mapping of vertices. With the same notation as before, we define $\hat{S}_t := \psi^{-1}(t)$ to be the set of vertices in $\hat{T}$ corresponding to terminal $t \in S$ (the *group* of t). The notion of group will be needed later to define a proper 2-GST instance.

We have all the ingredients for formulating our ILP. We define indicator variables $x_e \in \{0, 1\}$ for all $e \in E$, which take value $x_e = 1$ iff $e \in H$ (H is an optimal solution to 2-DST). The objective function that we wish to minimize is $\sum_{e \in E} c_e x_e$. Similarly, we define indicator variables $\hat{x}_e \in \{0, 1\}$ for all $e \in \hat{E}$, which take value $\hat{x}_e = 1$ iff $e \in E(\hat{H})$.

Now we define our constraints. First we define a set of linear constraints, denoted by LP_{gst}, which models the fact that, for each $t \in S$, $\hat{H}$ must contain two edge disjoint paths from $\hat{r}$ to the group $\hat{S}_t$. So, we introduce flow variables $\hat{f}^t_{\hat{e}} \in \{0, 1\}$ for all $\hat{e} \in \hat{E}$ and all terminals $t \in S$. The constraints LP_{gst} are given in Figure 1. We remark that LP_{gst} are the linear constraints of the standard LP relaxation for the 2-GST problem with the root $\hat{r}$ and groups $\hat{S}_t$ for $t \in S$ in which the underlying graph is a *tree*. This is a crucial part of our formulation because this LP has a large integrality gap on general graphs [47].

Next we define the set of constraints LP_{cong} that formulates (implicitly) a mapping $\phi : \hat{E} \to 2^E$ of edges $\hat{e} = \hat{u}\hat{v}$ of $\hat{T}$ into $\psi(\hat{u}), \psi(\hat{v})$ paths of G. We introduce the following new flow variables: $f_{\hat{e},e} \in \{0, 1\}$, for

Fig. 1 The LP_{gst} constraints.

$$\begin{array}{rcll} \hat{f}^t_{\hat{e}} & \leq & \hat{x}_{\hat{e}} & \forall \hat{e} \in \hat{E}, \forall t \in S \\ \sum_{\hat{e} \in \delta^{in}_{\hat{T}}(\hat{v})} \hat{f}^t_{\hat{e}} & = & \sum_{\hat{e} \in \delta^{out}_{\hat{T}}(\hat{v})} \hat{f}^t_{\hat{e}} & \forall t \in S, \forall \hat{v} \in \hat{V} - (\hat{S}_t \cup \{\hat{r}\}) \\ \sum_{\hat{e} \in \delta^{in}_{\hat{T}}(\hat{S}_t)} \hat{f}^t_{\hat{e}} & \geq & 2 & \forall t \in S \end{array}$$

all $\hat{e} \in \hat{E}$ and $e \in E$. Intuitively, the set of edges $e \in E$ with $f_{\hat{e},e} = 1$ form the path $\phi(\hat{e})$. Clearly one has $f_{\hat{e},e} \leq x_e$. In order to satisfy the bounded congestion property, we impose that, for a given $e \in E$, the sum of variables $f_{\hat{e},e}$ is upper bounded by $\beta \cdot x_e$, where $\beta = O(D|S|^{1/D})$ comes from the Height Reduction Theorem (Theorem 6). These LP constraints are given in Figure 2.

Fig. 2 The constraints LP_{cong}.

$$\begin{array}{rcll} f_{\hat{e},e} & \leq & x_e & \forall \hat{e} = \hat{u}\hat{v} \in \hat{E}, \forall e \in E \\ \sum_{e \in \delta^{out}_G(u), u = \psi(\hat{u})} f_{\hat{e},e} & = & \hat{x}_{\hat{e}} & \forall \hat{e} = \hat{u}\hat{v} \in \hat{E} \\ \sum_{e \in \delta^{in}_G(u), u = \psi(\hat{u})} f_{\hat{e},e} & = & 0 & \forall \hat{e} = \hat{u}\hat{v} \in \hat{E} \\ \sum_{e \in \delta^{in}_G(w)} f_{\hat{e},e} & = & \sum_{e \in \delta^{out}_G(w)} f_{\hat{e},e} & \forall \hat{e} = \hat{u}\hat{v} \in \hat{E}, \forall w \in V - \{\psi(\hat{u}), \psi(\hat{v})\} \\ \sum_{\hat{e} \in \hat{E}} f_{\hat{e},e} & \leq & \beta \cdot x_e & \forall e \in E \end{array}$$

It remains to enforce the divergency property. We introduce a final set of new variables: $f^t_{\hat{e},e} \in \{0,1\}$, for all $\hat{e} \in \hat{E}$, $e \in E$, and $t \in S$. Intuitively, the edges $e \in E$ with $f^t_{\hat{e},e} = 1$ indicate whether e is part of one of the two edge disjoint paths in H from r to t. In an integral solution, for a given $e \in E(H)$ and t, at most one $f^t_{\hat{e},e}$ can be set to 1. This guarantees that the mapping ϕ maps two $\hat{r}$-$\hat{S}_t$ edge-disjoint paths in the shallow tree into two edge disjoint paths in the original graph from r to t. The set of constraints LP_{div} is described in Figure 3.

By relaxing the integrality constraints on the variables, we obtain an LP relaxation LP-2-DST for 2-DST, presented in Figure 4.

4 Approximation Algorithm: Rounding via Tree Embedding

In this section, we present our approximation algorithm for 2-DST. Our algorithm starts by solving LP-2-DST. Denote by $\{x_e, \hat{x}_{\hat{e}}, \hat{f}^t_{\hat{e}}, f_{\hat{e},e}, f^t_{\hat{e},e}\}_{e \in E, \hat{e} \in \hat{E}, t \in S}$ an optimal fractional solution. We then execute for $O(D \log n)$ times a *rounding procedure*, consisting of two main steps: the *GST rounding* and the *path mapping*. The union of all the solutions obtained is the approximate solution, which is feasible w.h.p.

In more detail, consider a given iteration j. The variables $\{\hat{x}_{\hat{e}}\}_{\hat{e} \in \hat{E}}$ provide a feasible solution to the standard LP for 2-GST on trees. In the GST rounding step, we apply the rounding algorithm by Garg et al. [21], which we refer to as GKR rounding, to round these variables. This gives us a subtree $\hat{H}_j = (\hat{V}_j, \hat{E}_j)$ of $\hat{T}$.

Fig. 3 The constraints LP_{div}

$$
\begin{array}{rcll}
f^t_{\hat{e},e} & \leq & f_{\hat{e},e} & \forall e \in E, \forall \hat{e} \in \hat{E}, \forall t \in S \\
\displaystyle\sum_{e \in \delta^{out}_G(u), u=\psi(\hat{u})} f^t_{\hat{e},e} & = & \hat{f}^t_{\hat{e}} & \forall \hat{e} = \hat{u}\hat{v} \in \hat{E}, \forall t \in S \\
\displaystyle\sum_{e \in \delta^{in}_G(u), u=\psi(\hat{u})} f^t_{\hat{e},e} & = & 0 & \forall \hat{e} = \hat{u}\hat{v} \in \hat{E}, \forall t \in S \\
\displaystyle\sum_{e \in \delta^{in}_G(w)} f^t_{\hat{e},e} & = & \displaystyle\sum_{e \in \delta^{out}_G(w)} f^t_{\hat{e},e} & \forall \hat{e} = \hat{u}\hat{v} \in \hat{E}, \forall t \in S, \forall w \in V - \{\psi(\hat{u}), \psi(\hat{v})\} \\
\displaystyle\sum_{\hat{e} \in \hat{E}} f^t_{\hat{e},e} & \leq & x_e & \forall e \in E, \forall t \in S
\end{array}
$$

Fig. 4 LP relaxation LP-2-DST.

$$
\begin{array}{ll}
\min & \displaystyle\sum_{e \in E} c_e \, x_e \\
\text{s.t.} & LP_{gst} \\
& LP_{cong} \\
& LP_{div} \\
& 0 \leq x_e, \hat{x}_{\hat{e}}, f_{\hat{e},e}, \hat{f}^t_{\hat{e}}, f^t_{\hat{e},e} \leq 1 \quad \forall \hat{e} \in \hat{E}, \forall e \in E, \forall t \in S
\end{array}
$$

In the path mapping step, we consider each edge $\hat{e} = \hat{u}\hat{v} \in \hat{E}_j$, where $u = \psi(\hat{u})$ and $v = \psi(\hat{v})$. We randomly map $\hat{e}$ to a u, v-path in G. To this aim, we interpret variables $\{\hat{f}_{\hat{e},e}\}_{e \in E}$ as a distribution $\mathbb{P}_{\hat{e}}$ over u, v-paths, and we sample according to this distribution. We repeat this sampling $O(\beta \log D)$ many times to guarantee that we have the desired properties (which will be discussed later) with sufficiently large probability.

Our main algorithm is presented in Algorithm 1.

Algorithm 1 Round 2-DST

Solve LP-2-DST and obtain a fractional solution $\{x_e, \hat{x}_{\hat{e}}, \hat{f}^t_{\hat{e}}, f_{\hat{e},e}, f^t_{\hat{e},e}\}_{e \in E, \hat{e} \in \hat{E}, t \in S}$.
for $j = 1$ **to** $20D \ln n$ **do**
    Round variables $\hat{x}_{\hat{e}}$ using GKR Rounding, and obtain a subtree $\hat{H}_j = (\hat{V}_j, \hat{E}_j)$ of $\hat{T}$.
    for each $\hat{e} = \hat{u}\hat{v} \in \hat{E}_j$, $u = \psi(\hat{u})$ and $v = \psi(\hat{v})$ **do**
        for $\ell = 1$ **to** $(4\beta + 2) \ln D$ **do**
            Sample a u, v-path $P^{\hat{e}}_\ell$ in G from the distribution $\mathbb{P}_{\hat{e}}$.
        end for
    end for
    Let H_j be the union of all sampled paths
end for
return $H := \cup_j H_j$.

The GKR Rounding algorithm is discussed in Section 4.1. The construction of path distributions is presented in Section 4.2. We then analyze our algorithm in Section 4.3.

4.1 GKR Rounding

Let $\hat{T}$ be the shallow tree. We may think that each edge is directed from the root. Let $\rho(\hat{e})$ denote a parent of an edge $\hat{e} \in \hat{E}$, i.e., $\rho(\hat{e})$ is an edge adjacent to $\hat{e}$ that is closer to the root. Consider the constrains LP_{gst} on variables $\hat{x}_{\hat{e}}$ and $\hat{f}^t_{\hat{e}}$. This is indeed the standard LP for 2-GST. Hence, we can apply GKR rounding. Assume w.l.o.g. that $\hat{x}_{\hat{e}} \leq \hat{x}_{\rho(\hat{e})}$. GKR algorithm considers edges in order of increasing distance from the root. Each edge $\hat{e} = \hat{r}\hat{v}$ incident to the root $\hat{r}$ is marked independently with probability $\hat{x}_{\hat{e}}$. Any other edge $\hat{e} \in \hat{E}$ whose parent edge has been marked is marked independently with probability $\hat{x}_{\hat{e}}/\hat{x}_{\rho(\hat{e})}$. Each marked edge is added to the output tree. In our case, this gives the graph $\hat{H}_j$.

Next lemma summarizes the properties of GKR Rounding that we will need in the analysis.

Lemma 7 ([21, 44])**.** *Consider the run of GKR rounding algorithm on a D-shallow tree $\hat{T}$ with variables $\{\hat{x}_{\hat{e}}, \hat{f}^t_{\hat{e}}\}_{\hat{e}\in\hat{E},t\in S}$ given by a fractional solution to the standard GST LP. Let $\hat{H}$ be the solution sampled by the algorithm, $t \in S$, and $\mu_t := \sum_{\hat{e}\in\delta^{in}_{\hat{T}}(\hat{S}_t)} \hat{f}^t_{\hat{e}}$. Then*

$$\Pr[\hat{e} \in \hat{H}] = \hat{x}_{\hat{e}} \quad \textit{and} \quad \Pr[\hat{H} \textit{ contains an } \hat{r}, \hat{S}_t\textit{-path}] \geq \frac{\mu_t}{2D}.$$

4.2 Constructing Path Distributions

Now we discuss how to construct a path distribution $\mathbb{P}_{\hat{e}}$ on each edge $\hat{e} = \hat{u}\hat{v} \in \hat{E}$. Let $u = \psi(\hat{u})$ and $v = \psi(\hat{v})$. Observe that the variables $F = \{f_{\hat{e},e}\}_{e\in E}$ form a u, v-flow. Thus, we can decompose F into a collection of flow paths, say $\{f^{\hat{e}}_P\}_{P\in\mathcal{P}_{uv}}$, where $\mathcal{P}_{uv}$ is the set of all u, v-paths in G, so that

$$\sum_{P\in\mathcal{P}_{uv}:e\in E(P)} f^{\hat{e}}_P = f_{\hat{e},e}.$$

The value of the flow F is $\hat{x}_{\hat{e}} = \sum_{e\in\delta^{out}_G(u)} f_{\hat{e},e}$. Thus, $\{f^{\hat{e}}_P/\hat{x}_{\hat{e}}\}_{P\in\mathcal{P}_{uv}}$ gives a collection of flow paths whose total flows is one, and we can interpret this as a distribution over flow paths. This implies the following lemma.

Lemma 8. *Consider an edge $\hat{e} = \hat{u}\hat{v} \in \hat{E}$ and its corresponding variables $\{f_{\hat{e},e}\}_{e\in E}$. Let $u = \psi(\hat{u})$ and $v = \psi(\hat{v})$. Then there exists a distribution of u, v-paths $\mathbb{P}_{\hat{e}}$ such that for all $e \in E$:*

$$\Pr_{P\sim\mathbb{P}_{\hat{e}}}[e \in E(P)] = \frac{1}{\hat{x}_{\hat{e}}} \cdot f_{\hat{e},e}$$

4.3 Analysis

Next we analyze Algorithm 1. We start with the simpler part of our analysis, namely bounding the expected cost of H.

Lemma 9. *The expected cost of H is $O(D^3 h^{2/D} \log D \log n) \cdot \sum_{e\in E} c_e x_e$.*

Proof. Let us bound the expected cost of $H_j = (V_j, E_j)$. For each edge $e \in E$,

$$\begin{aligned}
\Pr[e \in E_j] \quad &\leq \quad \sum_{\hat{e}\in\hat{E}} \Pr\left[\hat{e} \in \hat{E}_j\right] \cdot \Pr\left[\left(e \in \bigcup_{\ell=1}^{(4\beta+2)\ln D} E\left(P^{\hat{e}}_\ell\right)\right) \middle| \hat{e} \in \hat{E}_j\right] \\
&\overset{\text{Lem. 7 and 8}}{\leq} \quad \sum_{\hat{e}\in\hat{E}} \hat{x}_{\hat{e}} \cdot O(\beta \log D) \cdot \frac{f_{\hat{e},e}}{\hat{x}_{\hat{e}}} \overset{\text{by } LP_{cong}}{\leq} O(\beta^2 \log D) \cdot x_e
\end{aligned}$$

Thus, the expected cost of H_j is $O(\beta^2 \log D)$ times the LP value. The claim follows since there are $O(D \log n)$ iterations and $\beta = O(Dh^{1/D})$. $\square$

We next show that our algorithm gives a feasible solution to 2-DST with high probability. This is the most complicated part of the analysis.

The initial part of our analysis resembles the analysis in [5] for k-GST. We prove feasibility using Menger's theorem and a cut argument. By Menger's theorem, the solution subgraph $H \subseteq G$ contains two edge disjoint r,t-paths if and only if $H - \{e\}$ contains an r,t-path for every edge $e \in E(H)$. Therefore, we will focus on a given such pair (e,t). Our goal is to show that our rounding algorithm buys with sufficiently high probability an r,t-path *not using* the edge e. For this purpose, we exploit the fact that, according to Lemma 7, if we reduce the flow associated to some group $\hat{S}_t$, GKR algorithm will still connect $\hat{r}$ to $\hat{S}_t$ with sufficiently large probability provided that the residual amount of flow μ'_t from $\hat{r}$ to $\hat{S}_t$ is large enough.

At this point, we might try to reduce the flow by the amount $f_{\hat{e},e}$ for each edge $\hat{e} \in \hat{E}$. One can show that μ'_t would remain large enough, but unfortunately this in not sufficient in our case. Indeed, we might still have a fairly high probability to use the edge e due to the probabilistic distribution over paths: for any given $\hat{e}$, the sampled path $P^{\hat{e}}_\ell$ contains e with probability $f_{\hat{e},e}/\hat{x}_{\hat{e}}$. So, we can "safely" use the edge $\hat{e}$ only if the complementary probability $(\hat{x}_{\hat{e}} - f_{\hat{e},e})/\hat{x}_{\hat{e}}$ is sufficiently large. We say that an edge of the latter type is good, and we wish to use only good edges.

Formally, we say that an edge $\hat{e} \in \hat{E}$ is *good* against e if

$$\hat{x}_{\hat{e}} - f_{\hat{e},e} \geq \frac{1}{2\beta} \cdot f_{\hat{e},e}.$$

Otherwise, we say that $\hat{e}$ is *bad* against e. If the edge e is clear from the context, we will simply say that $\hat{e}$ is bad (respectively, good). Similarly, we say that a path $\hat{P}$ in $\hat{T}$ is *good* (against e) if all edges of $\hat{P}$ are good. Otherwise, we say that $\hat{P}$ is bad.

We claim that we can route an $\hat{r},\hat{S}_t$-flow of value at least $1/2$ using only good edges and even after decreasing the capacity of edge $\hat{e}$ by $f_{\hat{e},e}$. In particular, we prove the following lemma.

Lemma 10. *Let $e \in E$ and $t \in S$. Let $\hat{E}_{bad} \subseteq \hat{E}$ be the subset of bad edges against e, and $\hat{E}_{good} = \hat{E} - \hat{E}_{bad}$. Consider the shallow tree $\hat{T}' = \hat{T} - \hat{E}_{bad}$ with capacities $\{\hat{x}'_{\hat{e}}\}_{\hat{e}\in\hat{E}}$, where $\hat{x}'_{\hat{e}} = \hat{x}_{\hat{e}} - f_{\hat{e},e}$ for all edges $\hat{e} \in \hat{T}$. Then $\hat{T}'$ supports an $\hat{r},\hat{S}_t$-flow of value at least* $1/2$*.*

Proof. Recall that $\{\hat{f}^t_{\hat{e}}\}_{\hat{e}\in\hat{E}}$ supports an $\hat{r},S_t$-flow of value at least 2 on $\hat{T}$. Thus, for any cut $\hat{U}$ that separates $\hat{r}$ and $\hat{S}_t$, (i.e., $\hat{U} \subseteq \hat{V}$, $\hat{r} \in \hat{U}$ and $\hat{S}_t \subseteq (\hat{V} - \hat{U})$), we must have

$$\sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})} \hat{f}^t_{\hat{e}} \geq 2.$$

We will prove later the following inequality

$$\hat{f}^t_{\hat{e}} - f^t_{\hat{e},e} \leq \hat{x}_{\hat{e}} - f_{\hat{e},e} \text{ for any edge } \hat{e} \in \hat{E}. \tag{1}$$

Now we consider the capacities of edges leaving $\hat{U}$ in the absence of bad edges.

$$
\begin{aligned}
\sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})\cap\hat{E}_{good}} \hat{x}'_{\hat{e}} &= \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})\cap\hat{E}_{good}} (\hat{x}_{\hat{e}} - f_{\hat{e},e}) = \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})} (\hat{x}_{\hat{e}} - f_{\hat{e},e}) - \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})\cap\hat{E}_{bad}} (\hat{x}_{\hat{e}} - f_{\hat{e},e}) \\
&\overset{\text{By (1)}}{\geq} \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})} (\hat{f}^t_{\hat{e}} - f^t_{\hat{e},e}) - \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})\cap\hat{E}_{bad}} (\hat{x}_{\hat{e}} - f_{\hat{e},e}) \\
&\overset{\text{by } LP_{gst}}{\geq} 2 - \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})} f^t_{\hat{e},e} - \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})\cap\hat{E}_{bad}} (\hat{x}_{\hat{e}} - f_{\hat{e},e}) \\
&\overset{\text{by def. bad}}{\geq} 2 - \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})} f^t_{\hat{e},e} - \frac{1}{2\beta} \sum_{\hat{e}\in\delta^{out}_{\hat{T}}(\hat{U})\cap\hat{E}_{bad}} f_{\hat{e},e} \geq 2 - \sum_{\hat{e}\in\hat{E}} f^t_{\hat{e},e} - \frac{1}{2\beta}\sum_{\hat{e}\in\hat{E}} f_{\hat{e},e} \\
&\overset{\text{by } LP_{div}}{\geq} 2 - x_e - \frac{1}{2\beta}\sum_{\hat{e}\in\hat{E}} f_{\hat{e},e} \overset{\text{by } LP_{cong}}{\geq} 2 - x_e - \frac{1}{2\beta}\cdot\beta x_e \overset{x_e\leq 1}{\geq} \frac{1}{2}
\end{aligned}
$$

Thus, by the Max-Flow-Min-Cut Theorem, the network $\hat{T}'$ with capacities $\{\hat{x}'_{\hat{e}}\}_{\hat{e}\in\hat{E}}$ supports an $\hat{r}, \hat{S}_t$-flow of value at least 1/2.

It remains to prove (1). The claim is trivially true if $f_{\hat{e},e} = 0$ since it implies $f^t_{\hat{e},e} = 0$. So, let us assume that e belongs to the support of $\{f_{\hat{e},e'}\}_{e'\in E}$. Again, we use the Max-Flow-Min-Cut Theorem. Consider $\hat{e} = \hat{u}\hat{v} \in \hat{E}$, and let $u = \psi(\hat{u})$ and $v = \psi(\hat{v})$. By the constraints of LP_{cong}, the graph G with capacities $\{f_{\hat{e},e'}\}_{e'\in E}$ supports a u, v-flow of value $\hat{x}_{\hat{e}}$. There must exist a minimum u, v-cut U^* that contains the edge e, provided that $f_{\hat{e},e} > 0$. To see this, observe that $\{f_{\hat{e},e'}\}_{e'\in E}$ induces a minimal flow network (as it is a flow itself), i.e., decreasing the capacity of any edge decreases the value of maximum flow by the same amount. So, every edge with positive capacity must contain in some minimum cut. Consequently, we have

$$
\begin{aligned}
\hat{x}_{\hat{e}} - f_{\hat{e},e} &= \left(\sum_{e'\in\delta^{out}_G(U^*)} f_{\hat{e},e'}\right) - f_{\hat{e},e} = \sum_{e'\in(\delta^{out}_G(U^*)-\{e\})} f_{\hat{e},e'} \\
&\overset{\text{By } LP_{div}}{\geq} \sum_{e'\in(\delta^{out}_G(U^*)-e)} f^t_{\hat{e},e'} = \left(\sum_{e'\in\delta^{out}_G(U^*)} f^t_{\hat{e},e'}\right) - f^t_{\hat{e},e} \geq \hat{f}^t_{\hat{e}} - f^t_{\hat{e},e}
\end{aligned}
$$

This completes the proof.

□

Next consider any good path against e in $\hat{T}$, say $\hat{P} \subseteq \hat{T}$, that connects $\hat{r}$ to a group $\hat{S}_t$. Then $\hat{P}$ maps to an r, t-path in the original graph G that does not use e with probability at least $1 - 1/D$.

Lemma 11. *Let $e \in E$ and $\hat{P} \subseteq \hat{T}$ be a good $\hat{r}, \hat{S}_t$-path against e. Suppose we map $\hat{P}$ to a subgraph $Q \subseteq G$ by sampling $(4\beta + 2)\ln D$ paths from the distribution $\mathbb{P}_{\hat{e}}$ for each $\hat{e} \in E(\hat{P})$. Then $Q - \{e\}$ contains an r, t-path with probability at least $1 - 1/D$.*

Proof. Consider an edge $\hat{e} = \hat{u}\hat{v} \in E(\hat{P})$. By Lemma 8, we have that any path P sampled from $\mathbb{P}_{\hat{e}}$ contains e with probability

$$
\Pr_{P\sim\mathbb{P}_{\hat{e}}}[e \in E(P)] \leq \frac{f_{\hat{e},e}}{\hat{x}_{\hat{e}}} \overset{def.good}{\leq} 1 - \frac{1}{2\beta + 1}
$$

Since we sample $(4\beta+2)\ln D$ paths from $\mathbb{P}_{\hat{a}}$, the probability that all the sampled paths contain e is at most

$$\Pr[\text{all paths } P \text{ sampled from } \mathbb{P}_{\hat{e}} \text{ contain } e] \leq \left(1-\frac{1}{2\beta+1}\right)^{2(2\beta+1)\ln D} \leq \left(\frac{1}{\mathbf{e}}\right)^{2\ln D} \leq \frac{1}{D^2}$$

(here $\mathbf{e}$ is the base of the natural logarithm.) We recall that $\hat{P}$ has length at most D. Thus, by the union bound,

$$\Pr[\exists \hat{e} \in E(\hat{P}) \text{ s.t. all the paths } P \text{ sampled from } \mathbb{P}_{\hat{e}} \text{ contain } e] \leq D \cdot \frac{1}{D^2} = \frac{1}{D}.$$

We conclude that, with probability at least $1-1/D$, for each $\hat{e} \in E(\hat{P})$ we sample at least one path in G that avoids e: the union of such avoiding paths forms a (possibly non-simple) r-t path that avoids e. $\square$

Now are ready to prove the feasibility of our solution H obtained from Algorithm 1.

Lemma 12. *The subgraph H returned from Algorithm 1 is a feasible 2-DST solution w.h.p.*

Proof. By Menger's theorem, H is a feasible 2-DST solution iff for every edge $e \in E$ and terminal $t \in S$, $H - \{e\}$ contains an r,t-path.

We claim that each subgraph $H_j - \{e\}$ contains an r,t-path with probability at least $1/(5D)$. First observe that, by Lemma 10, the capacities $\{\hat{x}'_{\hat{e}}\}_{\hat{e}\in\hat{E}}$ support an $\hat{r},\hat{S}_t$-flow of value at least $1/2$ through good paths. Thus, by Lemma 7, the GKR rounding algorithm guarantees that $\hat{H}_j$ contains a good $\hat{r},\hat{S}_t$-path with probability at least $1/(4D)$. Given the existence of a good path in $\hat{H}_j$, by Lemma 11, H_j contains an r,t-path avoiding e with probability at least $(1-1/D)$. Altogether, H_j contains such a path with probability at least $(1-1/D)/(4D) \geq 1/(5D)$.

Since H is a union of $20D\ln n$ subgraphs H_j's sampled independently, the probability that no subgraphs H_j contain an r,t-path is at most

$$\left(1-\frac{1}{5D}\right)^{20D\ln n} \leq \frac{1}{n^4}.$$

As we have at most n terminals and at most n^2 edges, it follows by the union bound that $H-\{e\}$ contains an r,t-paths, for every edge $e \in E$ and every terminal $t \in S$, with probability at least $1-1/n$. $\square$

5 Conclusions

We presented a non-trivial approximation algorithm for 2-DST on general graphs. Our approach crucially relies on a decomposition of a feasible solution into two divergent Steiner trees. It is known that an analogous decomposition does not exist for connectivity $k \geq 3$ [27, 3]. However, weaker decomposition theorems would be sufficient to exploit our basic approach. For example, is it possible to decompose a feasible solution to k-DST into a collection of $f(k) \cdot \mathrm{polylog}(n,h)$ trees so that, for each terminal $t \in S$ and for any edge-cut F of size $k-1$, there exists some tree in the collection that connects r to t using no edges from F? Such a result could be combined with our LP-rounding technique to achieve similar approximation ratios for any constant k. We are not aware of any such result nor of any counter-example. To support our conjecture, we show in Appendix B the existence of a weaker decomposition in undirected graphs that supports connectivity $\lceil k/2 \rceil$. Such decomposition allows us to design a bi-criteria approximation algorithm for a variant of k-GST, namely k-GST*, where all the k edge-disjoint paths must end at the same vertex.

Achieving a sub-polynomial approximation for 2-DST in polynomial time is another obvious open problem. However, this has been a major open problem for decades even for DST. On the positive side, it is likely that any future progress on DST can be extended to 2-DST via our approach.

Acknowledgment. We thank R. Ravi for suggesting the variant of k-GST.

References

[1] A. Agrawal, P. N. 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, 1995.

[2] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *FOCS*, pages 184–193, 1996.

[3] K. Bérczi and E. R. Kovács. A note on strongly edge-disjoint arborescences. In *Proceedings of the 7th Japanese-Hungarian Symposium on Discrete Mathematics and its Applications, June, 2011, Kyoto, Japan.*, pages 10–18, June 2011.

[4] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. Steiner tree approximation via iterative randomized rounding. *Journal of the ACM*, 60(1):6, 2013.

[5] P. Chalermsook, F. Grandoni, and B. Laekhanukit. On survivable set connectivity. In *SODA*, pages 25–36, 2015.

[6] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. *Journal of Algorithms*, 33(1):73–91, 1999.

[7] C. Chekuri and M. Pál. A recursive greedy algorithm for walks in directed graphs. In *FOCS*, pages 245–253, 2005.

[8] J. Cheriyan, B. Laekhanukit, G. Naves, and A. Vetta. Approximating rooted steiner networks. *ACM Transactions on Algorithms*, 11(2):8:1–8:22, 2014.

[9] J. Cheriyan and L. A. Végh. Approximating minimum-cost k-node connected subgraphs via independence-free graphs. *SIAM Journal on Computing*, 43(4):1342–1362, 2014.

[10] J. Cheriyan, S. Vempala, and A. Vetta. An approximation algorithm for the minimum-cost k-vertex connected subgraph. *SIAM J. Comput.*, 32(4):1050–1055, 2003. Preliminary version in STOC'02.

[11] J. Cheriyan and A. Vetta. Approximation algorithms for network design with metric costs. *SIAM Journal on Discrete Mathematics*, 21(3):612–636, 2007.

[12] G. Dahl. Directed steiner problems with connectivity constraints. *Discrete Applied Mathematics*, 47(2):109–128, 1993.

[13] J. Edmonds. Edge-disjoint branchings. *Combinatorial algorithms*, 9(91-96):2, 1973.

[14] J. Fakcharoenphol and B. Laekhanukit. An o(log$^{2k)}$-approximation algorithm for the k-vertex connected spanning subgraph problem. *SIAM Journal on Computing*, 41(5):1095–1109, 2012.

[15] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[16] M. Feldman, G. Kortsarz, and Z. Nutov. Improved approximation algorithms for directed Steiner forest. *Journal of Computer and System Sciences*, 78(1):279–292, 2012.

[17] L. Fleischer, K. Jain, and D. P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences*, 72(5):838–867, 2006.

[18] G. N. Frederickson and J. JáJá. Approximation algorithms for several graph augmentation problems. *SIAM J. Comput.*, 10(2):270–283, 1981.

[19] T. Fukunaga, Z. Nutov, and R. Ravi. Iterative rounding approximation algorithms for degree-bounded node-connectivity network design. *SIAM J. Comput.*, 44(5):1202–1229, 2015. Preliminary version in FOCS'12.

[20] H. N. Gabow. On the l_{infinity}-norm of extreme points for crossing supermodular directed network lps. *Math. Program.*, 110(1):111–144, 2007.

[21] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000.

[22] L. Georgiadis and R. E. Tarjan. Dominator tree certification and divergent spanning trees. *ACM Transactions on Algorithms*, 12(1):11, 2016. Preliminary version in SODA'05.

[23] M. X. Goemans, A. V. Goldberg, S. A. Plotkin, D. B. Shmoys, É. Tardos, and D. P. Williamson. Improved approximation algorithms for network design problems. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. 23-25 January 1994, Arlington, Virginia.*, pages 223–232, 1994.

[24] A. Gupta, R. Krishnaswamy, and R. Ravi. Tree embeddings for two-edge-connected network design. In *SODA*, pages 1521–1538, 2010.

[25] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *STOC*, pages 585–594, 2003.

[26] C. S. Helvig, G. Robins, and A. Zelikovsky. An improved approximation scheme for the group Steiner problem. *Networks*, 37(1):8–20, 2001.

[27] A. Huck. Disproof of a conjecture about independent branchings in k-connected directed graphs. *Journal of Graph Theory*, 20(2):235–239, 1995.

[28] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[29] R. Khandekar, G. Kortsarz, and Z. Nutov. Approximating fault-tolerant group-steiner problems. *Theorerical Computer Science*, 416:55–64, 2012.

[30] S. Khuller and B. Raghavachari. Improved approximation algorithms for uniform connectivity problems. *Journal of Algorithms*, 21(2):434–450, 1996.

[31] G. Kortsarz and Z. Nutov. Approximating k-node connected subgraphs via critical graphs. *SIAM J. Comput.*, 35(1):247–257, 2005. Preliminary version in STOC'04.

[32] G. Kortsarz and Z. Nutov. Approximating minimum-cost connectivity problems. In *Handbook of Approximation Algorithms and Metaheuristics.* Chapman & Hall/CRC, 2007.

[33] E. R. Kovács. Network coding. Master's thesis, Eötvös Loránd University, Budapest, 2007. (in Hungary).

[34] B. Laekhanukit. Parameters of two-prover-one-round game and the hardness of connectivity problems. In *SODA*, pages 1626–1643, 2014.

[35] B. Laekhanukit. An improved approximation algorithm for the minimum cost subset k-connected subgraph problem. *Algorithmica*, 72(3):714–733, 2015.

[36] B. Laekhanukit. Approximating directed Steiner problems via tree embedding. *ICALP*, pages 74:1–13, 2016.

[37] V. Melkonian and É. Tardos. Algorithms for a network design problem with crossing supermodular demands. *Networks*, 43(4):256–265, 2004.

[38] Z. Nutov. Approximability status of survivable network problems. Preprint available at `http://www.openu.ac.il/home/nutov/SN.pdf`.

[39] Z. Nutov. Approximability status of survivable network problems. Preprint available at `http://www.openu.ac.il/home/nutov/kCS.pdf`.

[40] Z. Nutov. Approximating minimum-cost connectivity problems via uncrossable bifamilies. *ACM Transactions on Algorithms*, 9(1):1, 2012.

[41] Z. Nutov. Approximating subset k-connectivity problems. *Journal of Discrete Algorithms*, 17:51–59, 2012.

[42] Z. Nutov. Degree constrained node-connectivity problems. *Algorithmica*, 70(2):340–364, 2014.

[43] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC*, pages 255–264, 2008.

[44] T. Rothvoß. Directed Steiner tree and the Lasserre hierarchy. *CoRR*, abs/1111.5473, 2011.

[45] K. Steiglitz, P. Weiner, and D. Kleitman. The design of minimum-cost survivable networks. *IEEE Transactions on Circuit Theory*, 16(4):455–460, 1969.

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

[47] L. Zosin and S. Khuller. On directed Steiner trees. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA.*, pages 59–63, 2002.

A A Reduction from 2-DSS to 2-DST

It is known that an approximation algorithm for k-DST implies an approximation algorithm for k-DSS for both edge and vertex connectivity versions [30, 35, 41]. The reductions of these two cases are slightly different, but they are based on the same technique.

Edge-Connectivity. We first consider the edge-connectivity version of 2-DSS and 2-DST. It is known that an α-approximation algorithm for k-DST yields an approximation algorithm for k-DSS with a loss of factor two [30]. To be precise, let G be an input graph of k-DSS and $S \subseteq V(G)$ be a set of terminals, and let $\mathcal{A}$ be an α-approximation algorithm for k-DST. We form an instance of k-DST by taking an arbitrary terminal $r \in S$ as a root vertex of k-DST and taking $S' = S - \{r\}$ as a set of terminals. Then we solve in-rooted-version and out-rooted-version of k-DST, separately, and take the union of the two solutions. Thus, every terminal $t \in S - \{r\}$ has k edge-disjoint paths to and from the root. It then follows by the transitivity of edge-connectivity that there are k edge-disjoint paths joining every pair of terminals. Therefore, this gives a 2α-approximation algorithm for k-DSS.

Vertex-Connectivity. Now, we consider the case of vertex-connectivity of k-DSS and k-DST. The reduction is more involved than the case of edge-connectivity since vertex-connectivity does not have the transitivity property. Here we need to pay an extra factor of k^2. In particular, as shown in [30, 35, 41], an α-approximation algorithm for k-DST implies $(2\alpha + k^2)$-approximation algorithm for k-DSS.

The reduction is as follows. Let G be an input graph of k-DSS and $S \subseteq V(G)$ be a set of terminals, and let $\mathcal{A}$ be an α-approximation algorithm for k-DST. First, we take any subset R of k vertices from S. The we apply any (efficient) min-cost k-flow algorithms on every pair of vertices in R and obtain an set of edges E'. We then form an instance of the vertex-connectivity version of k-DST by adding an auxiliary vertex r as a root and joining it to every vertex of R, and then taking $S' = S - R$ as a set of terminals. We apply the algorithm for k-DST for both in-version and out-version and then take the union of these solutions with E' (that we obtained from the min-cost k-flow algorithm). It is not hard to see that the cost of the solution is at most $(\alpha + k^2)$ times the optimum, and the feasibility can be verified using a cut-based argument. (See [35] for more details.)

B Bicriteria Approximation Algorithm for a Variant of k-GST

In this section, we present an application of our algorithm for 2-DST to a variant of k-GST proposed by Gupta et al. [24]. Recall that in k-GST we wish to find a min-cost subgraph H of a weighted undirected graph $G = (V, E)$ that has k edge-disjoint paths from a given root r to each *group* $S_t \subseteq V$, $t = 1, 2, \ldots, h$. In the mentioned variant, we require that all such paths end at the same vertex $s_t \in S_t$. We refer to this problem as k-GST*. Gupta et al. presented an $O(\log^3 n \log^h \log\log n)$ approximation for the case of 2-GST*, and the algorithm of Chalermsook et al.[5] gives a bicriteria approximation algorithm[4] that provides connectivity $\Omega(k/\log n)$. We present an alternative bicriteria approximation algorithm that outputs a solution with cost at most $O(k \cdot D^3 \log D \cdot h^{2/D} \cdot \log n)$ times the optimum and provides connectivity at least $\lceil k/2 \rceil$.

Our algorithm is now based on decomposing the optimal solution to k-GST* into a collection of k trees $T_1, \ldots, T_k$ such that for any set of edges F of size $\lceil k/2 \rceil - 1$ in G, there exists a tree T_i that contains no edge in F. The algorithm and analysis then follow along the same line as that for 2-DST. But, we need to run the

[4] In [5], the authors considered the standard version of k-GST, but the algorithm also works for the variant of k-GST*.

outer loop of the rounding procedure (Step 2 of Algorithm 1) for $O(kD \ln n)$ times instead of $O(D \ln n)$ because the number of edge-cuts that we have to consider is now $n^{O(k)}$. This incurs an extra factor of $O(k)$ in the approximation guarantee.

It remains to show that the above decomposition exists. Observe that an optimal solution H to k-GST forms a graph that is k-edge-connected on the set $S^* = \{r, s_1, \ldots, s_h\}$. Thus, if we replace each undirected edge $\{u, v\}$ by two directed edges uv and vu, then we have a directed graph $\hat{H}$ such that S^* is k (strongly) edge-connected on $\hat{H}$. We may apply a splitting-off theorem to get rid of all the Steiner vertices (vertices in $V(H) - S^*$), resulting in a directed k-edge-connected graph $\hat{H}'$ whose vertex-set is S^*. Then, by Edmonds' Disjoint Arborescence Packing Theorem [13], we know that $\hat{H}'$ contains k edge-disjoint (out) spanning arborescences $\hat{T}'_1, \ldots, \hat{T}'_k$. Mapping them back to the original graph H, we have a collection of k trees $\mathbb{T} = \{T_1, \ldots, T_k\}$ such that any undirected edge $\{u, v\}$ is contained in at most two trees in $\mathbb{T}$. So, for any set of $\lceil k/2 \rceil - 1$ edges $F \subseteq E(H)$, there must exist a tree $T_i \in \mathbb{T}$ that contains no edge of F and connects every terminal to the root. Therefore, we have the decomposition as claimed.

Note that this is an evidence that a weaker version of the decomposition theorem (Theorem 5) might exist. The decomposition implies the following theorem.

Theorem 13 (Bicriteria k-GST*)**.** *For any $D \in [\log_2 h]$, there exists a randomized approximation algorithm that runs in $n^{O(D)}$ time and outputs a feasible solution H to k-GST* with cost $O(k \cdot D^3 \log D \cdot h^{2/D} \cdot \log n)$ times the optimum and provides connectivity at least $\lceil k/2 \rceil$.*

Remark. For the case of 2-GST, our algorithm gives a "true" approximation algorithm for both 2-GST and 2-GST*. To see this, we split each (undirected) edge $\{u, v\}$ of the input graph into two directed edges uv and vu with the same cost. We then add a terminal s_t for each group $\hat{S}_t$ and joining each vertex $v \in \hat{S}_t$ to s_t by a (directed) edge vs_t with zero-cost. This reduces 2-DST to 2-GST, but we have a small issue that the composition in Theorem 5 may give trees T_1 and T_2 such that the corresponding two edge-disjoint r, s_t-paths P_1 and P_2 contain both uv and vu edges. However, we may use a stronger form of Theorem 5 where we additionally require that the paths P_1 and P_2 are *strongly* divergent, i.e., only one of uv and uv can be contained in $E(P_1) \cup E(P_2)$ [22, 33]. Our approximation guarantee matches the results in [24] (albeit, with worse running time).