Survivable Network Design Revisited: Group-Connectivity

Anonymous

Abstract

In the classical survivable network design problem (SNDP), we are given an undirected graph $G = (V, E)$ with costs on edges and a connectivity requirement $k(s, t)$ for each pair of vertices. The goal is to find a minimum-cost subgraph $H \subseteq G$ such that every pair (s, t) are connected by $k(s, t)$ edge or (openly) vertex disjoint paths, abbreviated as EC-SNDP and VC-SNDP, respectively. The seminal result of Jain [FOCS'98, Combinatorica'01] gives a 2-approximation algorithm for EC-SNDP, and a decade later, an $O(k^3 \log n)$-approximation algorithm for VC-SNDP, where k is the largest connectivity requirement, was discovered by Chuzhoy and Khanna [FOCS'09, Theory Comput.'12]. While there is a rich literature on point-to-point settings of SNDP, the viable case of connectivity between subsets is still relatively poorly understood.

This paper concerns the generalization of SNDP into the subset-to-subset setting, namely Group EC-SNDP. We develop the framework, which yields the first non-trivial (true) approximation algorithm for Group EC-SNDP. Previously, only a bicriteria approximation algorithm is known for Group EC-SNDP [Chalermsook, Grandoni, and Laekhanukit, SODA'15], and a true approximation algorithm is known only for the single-source variant with connectivity requirement $k(S, T) \in \{0, 1, 2\}$ [Gupta, Krishnaswamy, and Ravi, SODA'10; Khandekar, Kortsarz, and Nutov, FSTTCS'09 and Theor. Comput. Sci.'12].

1 Introduction

In the *survivable network design* problem (SNDP), we are given a graph $G = (V, E)$ with non-negative edge-costs $c : E \to \mathbb{R}_{\geqslant 0}$ and a connectivity requirement between each pair of vertices $k : V \times V \to \mathbb{Z}_{\geqslant 0}$. A pair of vertices (s, t) with $k(s, t) > 0$ is called a *demand-pair*, and a vertex s that has a positive demand toward some vertex is called a *terminal*. The goal in SNDP is to find a minimum-cost subgraph $H \subseteq G$ that has $k(s, t)$ edge (or vertex) disjoint paths connecting every demand-pair (s, t). As SNDP captures the design of communication networks that can operate under failure conditions, this problem has been a focus of attention for many decades since its initial study in the late '60s [SWK69]. There have been many variants of SNDP, e.g., edge-connectivity SNDP (EC-SNDP), vertex-connectivity SNDP (VC-SNDP), where the goal is to connect demand-pairs by edge-disjoint paths and (openly) vertex disjoint paths, respectively. A number of studies have been devoted to SNDP, culminating in the discovery of a 2-approximation algorithm for EC-SNDP by Jain [Jai01], and an $O(k^3 \log n)$-approximation algorithm for VC-SNDP by Chuzhoy and Khanna [CK12], where k is the maximum connectivity requirement.

While EC-SNDP and VC-SNDP are decent models that capture many difficulties in designing a highly reliable network, these models do not address network design beyond *point-to-point*. Specifically, the classical setting of SNDP concerns only the survivability of communication between pairs of nodes in a network. However, in many applications, e.g., multicasting over an overlay network, distributed data center, and global routing in VLSI design, communications are generally taken place between two groups of nodes rather than point-to-point. These relatively modern applications require the generalization of SNDP where survivability is required in the communication between two communities or, in other words, in the *group-to-group* setting. We call the later model the *group-connectivity survivable network design problem* or *Group-SNDP* (resp., Group EC-SNDP for edge-connectivity).

Similar to the point-to-point network design, where the models are derived from the *Steiner tree* problem. The basic building block of community-to-community network design lays on the classical *group Steiner tree* problem (GST), where we are given subsets of vertices, called *groups*, and the goal is to find a minimum-cost tree that spans all the groups, i.e., the tree must connect to at least one vertex from each group. The generalization of GST into the fault-tolerant settings, namely the *fault-tolerant GST* or *k-edge-connected* GST (k-EC-GST), have been studied in [KKN12, GKR10, CGL15], and the generalization into pairwise group-connectivity, namely Group-SNDP (a.k.a., *survivable set-connectivity*) has been studied in [CGL15]. These studies culminated in a polylogarithmic approximation algorithm for k-EC-GST when $k = 2$ [GKR10], and a bicriteria approximation algorithm for the general edge-connectivity demands of Group SNDP (Group EC-SNDP) [CGL15]. Nevertheless, to the best of our knowledge, there was no known "true" (non-trivial) approximation algorithm for neither Group EC-SNDP nor k-EC-GST for $k \geqslant 3$. On the negative side, Group EC-SNDP or even k-EC-GST is known to admits no k^σ-approximation algorithm, for some fixed constant $0 < \sigma < 1$ unless $\mathsf{P} = \mathsf{NP}$ [CGL15], and when k is very large the problem admits no $O(q/\log q)$-approximation algorithm unless $\mathsf{NP} = \mathsf{ZPP}$ [LCLZ22].

This paper revisits the generalization of survivable network design problem in the group-connectivity setting, namely Group EC-SNDP. To be formal, in the Group EC-SNDP we are given an undirected graph $G = (V, E)$ with non-negative edge-costs, a collection of pairs of subsets of vertices $(S_1, T_1), \ldots, (S_q, T_q) \subseteq V \times V$ with prescribed connectivity requirements $k_i \in \mathbb{Z}_{\geqslant 0}$ for $i = 1, 2, \ldots, q$. The goal is to find a minimum-cost subgraph $H \subseteq G$ that has k_i edge-disjoint paths between every pair of subsets (S_i, T_i).

We present the first non-trivial (true) approximation algorithm for Group EC-SNDP, thus solving a long-standing open problem in the area of network design. The approximation guarantee of our algorithm is $O(k^2 \log^6 n(k \log n + \log q)))$, where k is the largest connectivity requirement. In particular, it is $O(k^3 \log^7 n)$ when $q = n^{O(k)}$, which resemblances the best known approximation ratio of $O(k^3 \log n)$ for VC-SNDP upto the polylogarithmic term. Notice that the number of demand-pairs q can be super-polynomial in n, and in this case, when $q \geqslant n^{\omega(k)}$, the approximation ratio of our algorithm becomes $O(k^2 \log^6 n \log q)$.

1.1 Related Work

The study of survivable network design problems was initiated in the late '60s [SWK69]. Since then many variants and generalizations have been modeled to capture a wide range of situations, e.g., edge-connectivity (EC-SNDP), vertex-connectivity (VC-SNDP), and element-connectivity (Elem-SNDP), subset-to-subset connectivity (Group-SNDP) and connectivity in directed graphs (Directed-SNDP). Please see [KM05] and [Nut18] for references therein.

EC-SNDP is the most well-studied among the problems in the class of survivable network design. It has been extensively studied in the '90s (see, e.g., [WGMV93, GGP+94, Jai01]), culminating in the discovery of a 2-approximation algorithm via iterative rounding method in the breakthrough result of Jain [Jai01]. The same technique generalizes to Elem-SNDP in the work of Fleischer, Jain, and Williamson [FJW06], thus giving a factor-two approximation algorithm for this variant as well. In contrast to EC-SNDP and Elem-SNDP, researchers have been struggling in developing approximation algorithms for VC-SNDP. To the best of our knowledge, there is only one known non-trivial approximation algorithm for VC-SNDP, which was discovered decades later in the work of Chuzhoy and Khanna [CK12], giving an $O(k^3 \log n)$-approximation algorithm to the problem, where k is the largest connectivity requirement. One reason for the difficulty of VC-SNDP is due to the hardness derived from the Label-Cover problem. Assuming $\mathsf{NP} \nsubseteq \mathsf{DTIME}(n^{\mathrm{polylog}(n)})$, almost all reasonable approximation ratios (i.e., $2^{\log^{1-\epsilon} n}$, for $\epsilon > 0$) have been ruled out by the work of Kortsarz, Krauthgamer, and Lee [KKL04], and the approximation ratios independent of k are ruled out by the work of Chuzhoy, Khanna, and Chakraborty [CCK08]. A more refined hardness of approximation was later shown in [Lae14]. Nevertheless, a special case of VC-SNDP called k-connected (spanning) subgraph problem admits constant factor approximation ratios for almost every parameter [CV14, FNR15, Nut20] except in the large connectivity regime, which still have polylogarithmic factor [FL12, Nut12]. Please see the recent comprehensive survey on VC-SNDP by Nutov [Nut18].

On directed graphs, both EC-SNDP and VC-SNDP are equivalent as there are polynomial-time reductions from one to the other. Thus, we denote them simply by Directed-SNDP. The problem seems to be much more difficult as it admits almost no approximation ratios in n due to the work of Dodis and Khanna [DK99], which gives a reduction from the notorious Label-Cover problem to Directed-SNDP. In fact, the hardness result holds even when all the connectivity requirements are $\{0, 1\}$. The bounds have been improved in the subsequent works to $n^{o(1)}$ under Gap-ETH in the work of Dinur [Din16] and has also been refined in [CLNV14, Lae14, DM18, Man19, LCLZ22]. To date, we know that even in the special case of single-source k-connectivity, the problem is at least as hard as the Label-Cover problem [CLNV14], and in the very recent work, Liao, Chen, Laekhanukit and Zhang showed that Directed-SNDP admits no non-trivial approximation algorithms unless $\mathsf{NP} = \mathsf{ZPP}$. To be more precise, Directed-SNDP admits neither $o(q/\log q)$ nor $o(2^{k/2}/k)$ approximation algorithms unless $\mathsf{NP} = \mathsf{ZPP}$, where q is the number of (positive) demand-pairs. Assuming the *Strongish Planted Clique Hypothesis* [MRS21], even $o(q)$-approximation algorithm has been ruled out [LCLZ22]. Despite the difficulties, some special cases of Directed-SNDP admits a polylogarithmic approximation factor [CLWZ20, Nut21].

The Group SNDP is far less understood than other models for both edge and vertex connectivity variants. The study of the classical group Steiner tree problem is shown in [GKR00] and the generalization to the special case of single-source two-edge-connectivity was studied in [KKN12, GKR10], resulting in polylogarithmic approximation algorithms for these two cases. A polylogarithmic approximation algorithm for general demands is also known in a restricted setting of low treewidth graphs [CDE$^+$18]. However, prior to our paper, only a bi-criteria approximation algorithm is known for Group EC-SNDP on general graphs [CGL15]. Similar to Directed-SNDP, when the connectivity requirements are very large, Group EC-SNDP admits no non-trivial approximation. That is, no $o(q/\log q)$-approximation algorithm exists unless $\mathsf{NP} = \mathsf{ZPP}$, and no $o(q)$-approximation algorithm exists unless the Strongish Planted Clique Hypothesis is false [LCLZ22]. When focusing on the hardness ratio in terms of k, it is shown in [CGL15] (combined with [Lae14] and the improvement in [Man19]) that the approximation hardness is $k^{1/5-\epsilon}$, for $\epsilon > 0$, assuming $\mathsf{NP} = \mathsf{ZPP}$.

1.2 Result and Contribution

As mentioned earlier, the main result in this paper is the first non-trivial (true) approximation algorithm for Group EC-SNDP.

Theorem 1 (Main Result)**.** *The group edge-connectivity survivable network design problem admits a polynomial-time (randomized) $O(k^2 \log^6 n(k \log n + \log q))$-approximation algorithm, where k is the largest pairwise connectivity requirement. In particular, the approximation ratio becomes $O(k^3 \log^7 n)$ when $q \leqslant n^{O(k)}$.*

The key ingredient is a capacity-based probabilistic tree-embedding by Räcke [Räc08], which is also used in the previous work by Chalermsook, Grandoni and Laekhanukit [CGL15]. However, Chalermsook et al. were not able to derive a true approximation algorithm from the capacity-based tree-embedding because such a mapping is "lossy". More precisely, their algorithm is a randomized LP-rounding algorithm that runs on a tree distribution, which is an embedding of a fractional solution. However, due to the distortion, although there is an (S_i, T_i)-flow of value k_i in the tree distribution, when we map it back to the original graph, we can guarantee only a flow of value at least $\Omega(k/\beta)$, where $\beta = O(\log n)$ is the capacity-distortion of Räcke's tree-embedding. Consequently, the algorithm in [CGL15] is only guaranteed to output a solution with connectivity at least $\Omega(k/\log n)$.

To circumvent the issue of lossy embedding, we are required to invoke two techniques. First, we apply a weight-updating technique. Every time we add new edges to the partial solution, we update the weights (capacities) of the bought edges by scaling their capacities down so that the distortion on these edges has only mild effects toward distortion. However, the weight-update is still insufficient for us to reach the desired connectivity. This is because a cut consisting of bought edges alone may have its capacities scaled down too much so that the desired connectivity in the tree cannot be guaranteed. Thus, the connectivity issue remains and we need an additional ingredient. To this end, we observe that all the cuts in the graph are present in the embedded trees in the form of leaves-to-leaves paths. Hence, to reach the desired connectivity, we can simply connect all of them simultaneously whenever they are *connectivity deficient*. It is easier said than to be done as the number of tree-demand-pairs that we need to *cover* (that is, adding paths to satisfy the connectivity requirement) is, in general, exponential on the number of vertices. Thus, even if we have a logarithmic approximation algorithm for the underlying network design problem on trees (which is called *subset-connectivity* in [CGL15]), it can only give a polynomial approximation ratio. Nevertheless, we are able to show that the number of tree-demand-pairs needed to be covered is at most $q \cdot n^{2k} \cdot 2^{\mathrm{poly}(k)\beta}$, which is $qn^{\mathrm{poly}(k)}$ when $\beta = O(\log n)$. As a consequence, a polylogarithmic approximation algorithm for the subset-connectivity problem on trees in [CGL15] implies a poly(k)polylog(n) approximation algorithm for Group EC-SNDP.

The techniques that we developed yield a framework that turns a "lossy" probabilistic capacity mapping into a "lossless" network design algorithm. Hence, it can be applied to a more general setting whenever a lossy capacity-based probabilistic tree-embedding exists, provided that the distortion is $\beta < n^{1/3}$. The general form of our result is as follows.

Theorem 2 (Network Design via Lossy-Embedding)**.** *Suppose there exists a probabilistic capacity mapping that maps a capacitated directed or undirected n-vertex graph G into a distribution of tree such that*

- *The congestion in expectation is β;*

- *The height of all tree is $O(\log(nC))$, where C is the ratio of the largest to smallest capacity.*

Then there exists a randomized $O(\beta^2 k^2 \log^4 n(k\log n + k\beta + \log q))$ approximation algorithm, where k is the largest pairwise connectivity requirement and q is the number of demand-pairs.

In particular, if $\beta = O(1)$, then the ratio becomes $O(k^2 \log^4 n(k\log n + \log q))$. If $\beta = O(\log n)$, then the ratio becomes $O(k^2 \log^6 n(k\log n + \log q))$. If $\beta = O(n^c)$ where $c < 1/3$, then the ratio becomes $O(\beta^2 k^2 \log^4 n(k\beta + \log q))$.

Theorem 2 allows us to derive a slightly better approximation ratio on special classes of graphs, e.g., graphs with bounded pathwidth, bandwidth or cutwidth [BCE+20] and k-outer planar graphs [Eme11]. Note that here we exploit the equivalence between distance and capacity-based probabilistic tree-embedding observed by Andersen and Feige [AF09].

Theorem 3. *Consider the Group EC-SNDP. There exist polynomial-time (randomized) approximation algorithms with approximation ratios:*

- *$O(b^4 \cdot k^2 \log^4 n(k\log n + \log q))$ for Group EC-SNDP on graphs with b-bounded pathwidth (resp., bandwidth and cutwidth), where b is a constant.*
- *$O(c^{2k} \cdot k^2 \log^4 n(k\log n + kc^k + \log q))$ for Group EC-SNDP on k-outer planar graphs, where c is a universal constant.*

Lastly, we remark that a capacity-based probabilistic tree-embedding on *α-balanced directed graphs* exists as shown in the work of Ene, Miller, Pachocki and Sidford [EMPS16]. However, their work is pertained to the single-source congestion minimization problem, which is not clear whether the technique fits in our framework. If it is applicable, perhaps with some modification, then our framework will imply a non-trivial approximation algorithm for the *singe-source directed k-edge-connectivity* problem (also called the *k-edge-connected directed Steiner tree* problem in [GL17]), for all values of k, on α-balanced graphs.

2 Technique: Overview and Intuition

We first discuss the idea used in the work of Chalermsook, Grandoni, and Laekhanukit [CGL15]. The authors derived a bi-criteria approximation algorithm for Group EC-SNDP by first solving the standard LP-relaxation and then embedding the fractional solution into a probabilistic distribution of trees. They then iteratively round the fractional solution using the Garg-Konjevod-Ravi rounding algorithm for the group Steiner tree problem on trees [GKR00].

Due to the *congestion* (i.e., capacity-distortion) $\beta = O(\log n)$ of the Räcke's capacity-based probabilistic tree-embedding, the solution could only be guaranteed to reach a connectivity at least $\Omega(k/\log n)$. It is quite interesting that the problematic edges are those that have large LP-values, say $x_e \geqslant 1/\beta$. This is because the effect of distortion on the capacities of these edges are large. Surprisingly, the good case where we can reach the target connectivity is when all the edges in the optimal LP-solutions have *small* LP-values, i.e., $x_e < 1/\beta$.

It is very counterintuitive that we wish for an LP-solution with no large LP-values because these edges are supposed to be "good-to-have" as we can trivially round x_e to one and simply pay a factor $O(\log n)$ in the approximation ratio. This suggests the mix of *tree-rounding* and *trivial-rounding*. However, a straightforward approach would inevitably fail because the connectivity has already been lost in the tree distribution. To be specific, let us consider a cut $(X, V\backslash X)$ with capacity exactly k, i.e., a *tight* cut that separates a demand-pair (S_i, T_i). Suppose there are k/β edges with capacity 1 that have congestion β crossing $(X, V\backslash X)$. Then this cut will appear to have a flow of values k in the tree distribution. However, even if we buy all the (S_i, T_i)-paths in the support of the distribution, it would form only $O(k/\beta)$ edge-disjoint paths in the original graph.

To circumvent this issue, we study the effect of distortion of large-capacity edges and analyze in detail which cut will be capacity-deficit when we embed it to the tree distribution. To simplify the discussion, fix an edge-set F of $\ell \leqslant k-1$ edges in the graph. If a demand-pair (S_i, T_i) is not yet k-edge-connected in the current partial solution, then removing F from the solution subgraph will disconnect them. That is, F is a certificate that the graph has not yet reached the desired connectivity. The same applies to the tree-embedding. If we remove edges in the embedded-tree corresponding to F (i.e., an edge that maps to a path containing an edge in F) and the pair S_i and T_i are disconnected, then S_i and T_i are not k-edge-connected in the original graph even if we buy all the (S_i, T_i)-paths in the tree distribution. Now, observe that any edge with capacity (which is its LP-value) less than $1/4\ell\beta$ will have no effect in separating S_i from T_i because even if it has capacity-distortion β, and we remove k of them, it cannot possibly separate S_i from T_i in the tree-embedding. Hence, we may assume the edges in the graph have capacities at most $1/4\ell\beta$ by simply scaling down the capacities of edges with *large* LP-values, or more precisely, we cap the capacity of any edge to be at most $1/4\ell\beta$.

Next, we observe the embedded tree after removing edges corresponding to the set F of ℓ edges. One can see that a cut that has a lot of edges with *small* LP-values will not be disconnected by removing F. More precisely, if the total capacities of edges with small LP-values is at least 1, then it will have a capacity of at least 1/2 after removing F, and such a condition holds in every feasible LP-solution. As such, the cuts that are "shattered" in the tree-embedding are those that are already ℓ-edge-connected in the solution subgraph.

This observation is quite interesting for us as because it means that we can view a pair of shattered components as super nodes and ask only for edge-disjoint paths connecting these components. Towards this goal, we define new tree-demand-pairs based on the cuts and the embedding tree that are *shattered-free*. Then it is not hard to see that joining these subset pairs through edge-disjoint paths will imply the connectivity in the original graph. The last obstacle lies on the fact that the number of tree-demand-pairs can be as large as 2^n in general. The critical point in our argument is in bounding the number of "shattered-free" tree-demand-pairs and showing that every tree-demand-pair has enough flow for the rounding algorithm. Once these are all settled, our algorithm is guaranteed to connect all the tree-demand-pairs within the claimed bound.

3 Preliminary

Group Edge-Connectivity Survivable Network Design. The group edge-connected survivable network design problem (Group EC-SNDP) is defined as follows: Given an undirected graph $G = (V, E)$ with non-negative edge costs $c : E \to \mathbb{R}_{\geqslant 0}$, and a collection of q demand-pairs $(S_1, T_1), \ldots, (S_q, T_q)$ with a connectivity requirement $k_i \in \mathbb{Z}_{\geqslant 0}$ for each demand-pair (S_i, T_i), the goal is to find a minimum-cost subgraph $H \subseteq G$ such that H has k_i-edge-disjoint paths from S_i to T_i for every $i \in [q]$.

One may assume w.l.o.g. that the connectivity requirements are uniform. To see this, let $k = \max_{i\in[q]} k_i$. We add k auxiliary edges with zero-cost to the graph, say $(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)$.

Then, for each demand-pair (S_i, T_i) with requirement $k_i < k$, we add a_j to S_j and b_j to T_j for $j = 1, 2, \ldots, k - k_i$. This adds $k - k_i$ independent edges joining S_i and T_i and only increases the size of the instance by k even when q is superpolynomial in n. It is not hard to see that any feasible solution to the modified graph with uniform requirement k induces a feasible solution to the original graph with the same cost and vice versa. As such, our discussion will focus on the uniform case.

Sometimes we consider more than one graph and, to clarify the notations, we use $V(\cdot)$ or $E(\cdot)$ to mean the vertex and edge set of a graph, respectively. For a set $X \subseteq V$ of vertices, we denote by $\delta_G(X)$ the edge set between X and $V \backslash X$. The standard cut-based LP-relaxation of Group EC-SNDP is as follows. For notational convenience, let $x(F) = \sum_{e \in F} x_e$ for a set of edges $F \subseteq E$.

$$\begin{aligned} \min \quad & \sum_{e \in E(G)} c_e x_e & \\ \text{s.t.} \quad & \sum_{e \in \delta_G(X)} x_e \geqslant k & \forall i \in [q], \forall X : T_i \subseteq X \subseteq V \backslash S_i \\ & 0 \leqslant x_e \leqslant 1 & \forall e \in E(G) \end{aligned} \tag{1}$$

3.1 Räcke's Probabilistic Tree-Embedding

Our algorithm uses a tree-embedding with congestion β as a black box. The main result is proved by the fact that Räcke's probabilistic tree-embedding has congestion $\beta = O(\log n)$ in expectation. In this section, we give the formal definition of Räcke's probabilistic tree embedding. Let $G = (V, E)$ with capacity $\tilde{x} : E \to \mathbb{R}_{\geqslant 0}$. Denote a tree embedding of $(G, \tilde{x})$ by $(\mathcal{T}, \mathcal{M}, y)$, where $\mathcal{T}$ is a tree, $\mathcal{M}$ is a mapping from $V(\mathcal{T}) \bigcup E(\mathcal{T})$ to $V(G) \bigcup 2^{E(G)}$, and y is the corresponding capacity function on the edges of $\mathcal{T}$. For each node v of $\mathcal{T}$, $\mathcal{M}(v)$ is some vertex in G. Particularly, $\mathcal{M}$ induces a one-to-one mapping between the leaves of $\mathcal{T}$ and the vertices of G. For each edge $f = (u, v)$ in $\mathcal{T}$, its capacity is $y(f) = \sum_{e \in \delta_G(X)} \tilde{x}_e$, where $(X, V \backslash X)$ is a partition induced by the leaves of $\mathcal{T} - f$. In addition, $\mathcal{M}$ maps f to a path in G between $\mathcal{M}(u)$ and $\mathcal{M}(v)$.

We will use following notations in our analysis repeatedly. For each vertex v of G, $\mathcal{M}^{-1}(v)$ is the corresponding leaf of $\mathcal{T}$. Let $\mathcal{M}^{-1}(X) = \bigcup_{v \in X} \mathcal{M}^{-1}(v)$ for $X \subseteq V(G)$. For each edge e in G, let $\mathcal{M}^{-1}(e) = \{f \in E(\mathcal{T}) : e \in \mathcal{M}(f)\}$ and let $\mathcal{M}^{-1}(F) = \bigcup_{e \in F} \mathcal{M}^{-1}(e)$ for $F \subseteq E(G)$.

Congestion. For each edge $e \in E(G)$, the load of e on a tree $\mathcal{T} \in \mathcal{D}$ is defined as the sum of the capacities of edges $\mathcal{M}^{-1}(e)$, i.e., $\mathsf{load}(e) = \sum_{f \in \mathcal{M}^{-1}(e)} y(f)$. Let the relative load be $\mathsf{rload}(e) = \mathsf{load}(e) / \tilde{x}_e$. For a tree embedding $(\mathcal{T}, \mathcal{M}, y)$, the *congestion* of G is $\max_{e \in E(G)} \{\mathsf{rload}(e)\}$. Finally, we define β to be the expected congestion:

$$\beta := \max_{e \in E(G)} \mathbb{E}_{\mathcal{T} \sim \mathcal{D}}[\mathsf{rload}(e)],$$

where $\mathcal{D}$ is the probability distribution on trees given by Räcke's tree-embedding.

Theorem 4 ([Räc08]). *There exists a probabilistic embedding of an n-vertex graph G with edge capacities into a tree with expected congestion at most $\beta = O(\log n)$.*

The embedding also guarantees some properties about flows. Let $\mathsf{flow}_G^{\tilde{x}}(A, B)$ denote the maximum flow from $A \subseteq V(G)$ to $B \subseteq V(G)$ in G under capacity $\tilde{x}$. For any pair of disjoint subsets

$A, B \subseteq V(G)$, the value of the maximum flow between A and B in $\mathcal{T} \in \mathcal{D}$ is at least that of the maximum flow between A and B in G, i.e., $\mathsf{flow}_G^{\tilde{x}}(A,B) \leqslant \mathsf{flow}_{\mathcal{T}}^{y}(A,B)$. Since each edge has congestion at most β in expectation, any flow that can be routed on the tree distribution can be routed in the original graph with a loss of a factor of β, i.e., $\mathsf{flow}_G^{\tilde{x}}(A,B) \geqslant \frac{1}{\beta} \cdot \mathbb{E}_{\mathcal{T}\sim\mathcal{D}}[\mathsf{flow}_{\mathcal{T}}^{y}(A,B)]$.

It was also known that the height of the tree in Räcke's tree distribution can be bounded by the largest to smallest ratio of the capacity, and the number of trees in the support of the Racke's tree distribution is at most on $O(\mathrm{poly}(n))$.

Lemma 5 ([Räc08]). *All the trees $\mathcal{T}$ in the support of the Räcke's tree distribution have height $O(\log(nC))$, where C is the ratio of the largest to smallest capacity in $\tilde{x}$.*

4 Algorithm

We solve the LP in (1) and obtain the optimal LP solution $\{x_e\}_{e\in E(G)}$. Let $\ell = k-1$. The algorithm will construct a $(\ell+1)$-connected subgraph $H_{\ell+1}$.

Let β be the congestion parameter of the tree-embedding. We define two subsets of edges based on β and ℓ:

$$\mathsf{LARGE} = \left\{ e \in E(G) : x_e \geqslant \frac{1}{4\ell\beta} \right\},$$
$$\mathsf{SMALL} = \left\{ e \in E(G) : x_e < \frac{1}{4\ell\beta} \right\}.$$

Our algorithm buys edges in two different rounds. In the first step, we directly buy all edges in LARGE into the solution subgraph $H_{\ell+1}$. This incurs a factor $O(\ell\beta) = O(k\log n)$ in the approximation ratio.

After that, if $H_{\ell+1}$ is already $(\ell+1)$-connected, i.e., there are $\ell+1$ edge-disjoint paths from every S_i to T_i, then we are done. If not, we continue to run a dependent rounding algorithm on the tree-embedding. We build the Räcke's tree (a distribution of trees) with the capacity defined as follows. We will cap the capacity of the edges in LARGE to be exactly $\frac{1}{4\ell\beta}$, while keeping the same value for edges e with $x_e < \frac{1}{4\ell\beta}$. We also omit the fractional solutions that are smaller than $\frac{1}{2n^2} \cdot \frac{1}{4\ell\beta}$, to control the height of the trees. Formally, we set the capacity of each edge $e \in E(G)$ as

$$\tilde{x}_e = \begin{cases} \frac{1}{4\ell\beta} & \text{if } e \text{ is in } \mathsf{LARGE}, \\ 0 & x_e < \frac{1}{2n^2} \cdot \frac{1}{4\ell\beta}, \\ x_e & \text{otherwise.} \end{cases} \tag{2}$$

By the setup of $\tilde{x}$, the height of tree is bounded by $O(\log n)$.

Lemma 6. *Let $\mathcal{D}$ be the Räcke's tree distribution for $(G, \tilde{x})$, the height of each $\mathcal{T} \sim \mathcal{D}$ is at most $O(\log n)$.*

Proof. It directly follows from Lemma 5 since the largest to smallest ratio w.r.t. $\tilde{x}$ is $C \leqslant 2n^2$. □

Now we are ready to state our algorithm. (See Algorithm 1.) The algorithm first computes an LP solution and set up a basic solution subgraph $H_{\ell+1} = (V, E_{\ell+1})$ by setting $E_{\ell+1} = \mathsf{LARGE}$. Then it tries to add edges to $E_{\ell+1}$ to make $H_{\ell+1}$ $(\ell+1)$-connected in τ_1 rounds. In each round, it samples a tree from Räcke's tree distribution. Then it keeps sampling (in τ_2 rounds) certain edges in the tree by the rounding algorithm from Grandoni-Chalermsook-Laekhanukit [CGL15] and appends the corresponding edges in the graph G of the sampled edges on the tree to $E_{\ell+1}$.

We will present the main flow of the proof in Section 5. Also, for completeness, we will formally present the subroutine of tree rounding and its analysis in Section 6.

Algorithm 1 The Algorithm for Constructing The $(\ell+1=k)$-connected Subgraph $H_{\ell+1}$

```
Solve $x$ to be the fractional solution to the LP in (1).
Buy LARGE edge set where $x_e \geqslant \frac{1}{4\ell\beta}$ to $H_{\ell+1}$. (We use $H = (V, \mathsf{LARGE})$ to denote the current subgraph).
Set the capacity $\widetilde{x}$ of edges following Eqn (2).
Compute Räcke's tree distribution $\mathcal{D}$ for $(G, \widetilde{x})$.
for $\tau_1$ rounds do
    Sample a tree $(\mathcal{T}, \mathcal{M}, y)$ from $\mathcal{D}$.
    for $\tau_2$ rounds do
        Append TreeRounding $(G, \mathcal{T})$ to $H_{\ell+1}$.
return $H_{\ell+1}$.
```

5 Analysis: Component-level Paths and $(\ell+1)$-Connectivity

CGL's Tree Rounding as a Black Box Let us discuss the CGL's tree rounding subroutine in detail. The subroutine keeps sampling trees from the distribution. In each iteration, it randomly buys some edges in the tree with expected cost of $O(\text{polylog}(n)) \cdot \sum_{e \in E} c(e)x_e$. As a result, each path in the tree that carries one unit of flow (or at least some fixed constant) has a constant probability to be selected. Then if we have ψ number of demanding pairs with constant flow to connect. We need to suffer $O(\log(\psi))$ rounds to connect all of them in constant probability, which concludes in the $O(\text{polylog}(n) \cdot \log(\psi))$ approximation ratio. To be more precise, we will need the following lemma in our proof. For completeness, we leave the proof of the lemma in Section 6.

Lemma 7 (CGL's Tree Rounding)**.** *Suppose that there is a tree $\mathcal{T}$ with $\mathsf{height}(\mathcal{T}) = O(\log n)$ and an edge set $E' \subseteq E(\mathcal{T})$ that supports a flow of value at least f between two vertex sets A and B. If only edges from E' can be selected by TreeRounding$(G, \mathcal{T})$, then it connects A to B with constant probability ϕ with cost of $O(\frac{1}{f} \cdot \beta \cdot \log^3 n) \cdot \sum_{e \in E} c_e x_e$.*

In this part, we illustrate the big picture of our analysis. To show our main idea clearly, we assume that there is no edge with $x_e < \frac{1}{2n^2} \cdot \frac{1}{4\ell\beta}$. Although we need to scale down those tiny edges to control the height of Räcke's tree, it only incurs a constant factor loss of the approximation ratio, and we will consider it in the complete proof in the following subsections.

The goal of our analysis is to formulate a sufficient condition for the $\ell+1$ connectivity, which consists of a small number of demanding pairs (A, B) so that we can connect all of them with not too many iterations. In Algorithm 1, after buying LARGE edges, we have a subgraph $H_{\ell+1}$ that is not yet $(\ell+1)$-connected. Recall that we use $H = (V, \mathsf{LARGE})$ to mean the state of $H_{\ell+1}$ at that moment. Because H is not $\ell+1$-connected, there are some edge-sets $F \subseteq E(H)$ of ℓ edges such that $H \backslash F$ has no path from S_i to T_i for some i. To show that finally $H_{\ell+1}$ is $(\ell+1)$-connectivity, we need to prove for any edge-set F of size ℓ, there is a path in $H_{\ell+1}\backslash F$ for those unconnected pairs (S_i, T_i) in $H\backslash F$.

A tree $\mathcal{T}$ is *good* for the cut F if the load of F on $\mathcal{T}$ is at most $1/2$, i.e., $y(\mathcal{M}^{-1}(F)) \leqslant 1/2$. We prove that there is at least a tree in the support of the tree distribution that is *good* for F. For a fixed F, we only focus on a *good* tree because F can at most block a flow of $1/2$ in $\mathcal{T}$ and we does not lose too much connectivity in $\mathcal{T}$ if we remove F.

Lemma 8. *For each edge-set $F \subseteq E(H)$ of exactly ℓ edges, a tree sampled in $\mathcal{T} \sim \mathcal{D}$ is good for F with probability at least $1/2$.*

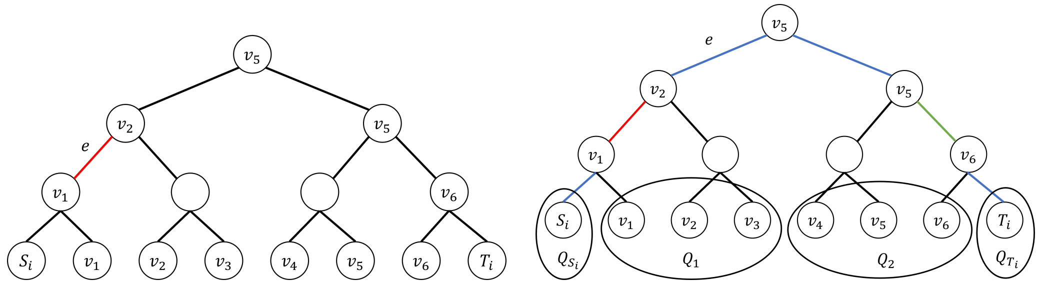
Proof. Since all edges in $F \subseteq E(H_\ell)$ are LARGE and $|F| = \ell$, the total capacity of F w.r.t $\tilde{x}$ is $\ell \cdot \frac{1}{4\ell\beta} = \frac{1}{4\beta}$. Note that the expected congestion of every edge in F is at most β by definition, so $\sum_{e\in F} \mathsf{load}(e) \leqslant 1/4$. We have $\mathbb{E}[y(\mathcal{M}^{-1}(F))] = \mathbb{E}[\sum_{e\in F} \mathsf{load}(e)] \leqslant 1/4$. By Markov's inequality, we have that

$$\begin{aligned}\mathbf{Pr}\left[\bar{\mathcal{E}}\right] &= \mathbf{Pr}\left[y(\mathcal{M}^{-1}(F)) \geqslant 1/2\right] \\ &\leqslant \mathbf{Pr}\left[y(\mathcal{M}^{-1}(F)) \geqslant 2\mathbb{E}[y(\mathcal{M}^{-1}(F))]\right] \leqslant 1/2.\end{aligned}$$

□

The naïve plan is to prove that the TreeRounding subroutine can return a new path avoiding F between every (S_i, T_i) with constant probability directly by Lemma 7. Hence, we need to show that the flow in $\mathcal{T}\backslash\mathcal{M}^{-1}(F)$ between S_i and $\mathcal{T}_i$ is at least a constant. Then we are done with an approximation ratio of $O((\ell \cdot \log n + \log q) \cdot \log^4 n)$ because we only have $O(qn^{2\ell})$ choices of (F, i) pair. The flow between S_i and T_i in $\mathcal{T}\backslash\mathcal{M}^{-1}(F)$ is at least its flow in $\mathcal{T}$ minus the value of $\mathcal{M}^{-1}(F)$. If we focus on a good tree $\mathcal{T}$, because F only blocks a flow of value $1/2$, it suffices to prove the total flow from S_i to T_i in $\mathcal{T}$ is larger than $1/2$. However, it is not guaranteed because the connectivity from S_i to T_i may totally lose if we remove F. Let us check the following example.

- (Refer to Figure 1(a)) The corresponding path $\mathcal{M}(e)$ in G of the red edge $e \in \mathcal{T}$ is part of a path from S_i to T_i in G. If there is some edge $f \in F$ such that $f \in \mathcal{M}(e)$ then the flow from S_i to T_i in $T\backslash\mathcal{M}^{-1}(F)$ becomes zero (disconnected). Even if $\mathcal{T}$ is a *good* tree, this situation can happen. By the feasibility of x we have $x(\delta_G(S_i \bigcup\{v_1\})) \geqslant \ell + 1$. However, if $\delta_G(S_i \bigcup\{v_1\}) \subseteq H_\ell$, $\tilde{x}(\delta_G(S_i \bigcup\{v_1\}))$ can be scaled down to $\frac{1}{4\beta} < 1/2$. Since F is allowed to block a flow of $1/2$ even in a good tree, it is possible to have some edge $f \in F$ such that $f \in \mathcal{M}(e)$.



(a) The red edge e contains an edge in F.

(b) A component-level path in blue. Red and green edges are crossing a component.

Figure 1: A typical case for illustrating the analysis.

To overcome the problem, we observe that the augmentation problem does not ask for a completely new path from S_i to T_i. Let us move into a component-level viewpoint. The edge-set F partitions the graph $H \backslash F$ into connected components. We know that every component Q is already $(\ell+1)$-connected in H so we do not need any new path inside Q. Let Q_{S_i} the union of components that intersect with S_i, and similarly Q_{T_i}. It means that a component-level path by edges in $G \backslash F$ that connects Q_{S_i} and Q_{T_i} suffices to show (S_i, T_i) is $(\ell+1)$-connected. As a counterpart of the previous bad case, we present the reason why finding a component-level path is possible by the following example.

- (Refer to Figure 1(b)) For the blue edge e in Figure 1(b), we have $\widetilde{x}_e \geqslant 1$ because $\mathcal{M}(e)$ is a component-level path and it is in the unique path in the tree from Q_{S_i} to Q_{T_i}. In this typical case, it is impossible to include any $f \in F$ in $M(e)$ in a good tree $\mathcal{T}$ because otherwise it would block a unit-value (larger than 1/2) flow.

The component-level path can be phrased in terms of the cut-based definition of tree-demand-pairs. Let $\mathbb{Q}$ be the set of connected components in $H \backslash F$. It is the same to say we need to connect every possible tree-demand-pairs defined as follows.

$$\begin{aligned} Z = \{(A, B) : (A, B) &\text{ is a partition of } \mathbb{Q} \\ &\text{such that } Q_{S_i} \in A \text{ and } Q_{T_i} \in B\} \end{aligned}$$

However, this straightforward definition creates 2^n number of tree-demand-pairs, deriving an approximation ratio of $O(n)$. We discover an interesting technique to bound both the number and the flow of tree-demand-pairs.

Connectivity by Shattered-Component-Level Paths. We observe that the reason why we need the help of the component-level path is because of the existence of the red edge, which makes a component no longer connected on the tree if F is removed. Referring to Figure 1(b), we call Q_1 a shattered component because of the red tree edge. On the contrary, assuming that the green tree edge does not include any edge in F, we call Q_2 an intact component. Because we can use the green edge to connect Q_{S_i} and Q_{T_i}, we do not need to view Q_2 as an intermediate component in the component-level path. We consider a new component-level path that only contains shattered components, and we call it *shattered-component-level paths*. Defining cut-based tree-demand-pairs on *shattered-component-level paths* (Refer to Section 5.1.) significantly decreases the number of tree-demand-pairs while keeping the lower bound of flows. In conclusion, we prove the flowing properties in Section 5.1, Section 5.2, and Section 5.3. We remark that Figure 1 only illustrates a special case when S_i and T_i contain only one vertex. In general cases, we may have different paths between S_i and T_i on the tree. The edges between a vertex cut may not only be one single blue edge or green edge. We will formally discuss them later.

1. (Refer to Section 5.1) If $(\mathcal{T}, \mathcal{M}, y)$ is good for F, we define the cut-based tree-demand-pairs by considering all possible shattered-component-level paths between all (S_i, T_i).

2. (Refer to Lemma 11) For each edge-set F, if we have sampled a good tree $\mathcal{T}$ for F and a subset of edges $E_b \subseteq E(\mathcal{T}) \backslash \mathcal{M}^{-1}(F)$ such that all tree-demand-pairs is connected in E_b, then there is a path from S_i to T_i in $(E_\ell \bigcup \mathcal{M}(E_b)) \backslash F$ for each demand-pair (S_i, T_i).

3. (Refer to Lemma 14) The flow in $\mathcal{T} \backslash \mathcal{M}^{-1}(F)$ between each tree-demand-pair is at least $\frac{1}{4\ell\beta}$.
4. (Refer to Lemma 16) Fix a tree $\mathcal{T}$. The number of tree-demand-pairs defined by all F and (S_i, T_i) is at most $qn^{2\ell} \cdot 2^{2\ell\beta}$. Note that it is $qn^{O(\ell)}$ if $\beta = O(\log n)$.

Given the properties as above, we can show that our algorithm achieves the claimed approximation ratio.

Lemma 9 (Connectivity). *Set $\tau_1 = O(\ell \log n)$ and $\tau_2 = O(\log q + \ell(\log n + \beta))$. The output graph $H_{\ell+1}$ of Algorithm 1 is $(\ell+1)$-connected for each demand-pair (S_i, T_i) with probability at least $1/2$.*

Proof. If we can prove that $H_{\ell+1} \backslash F$ still connects S_i with T_i for all F with $|F| = \ell$, and for all $i = 1, ..., q$, then $H_{\ell+1}$ is $(\ell+1)$-connected. Since every $\mathcal{T}$ in the support of D has $\mathsf{height}(\mathcal{T}) = O(\log n)$ by Lemma 6, combining with lemma 7, we have a constant probability ϕ to connect a tree-demand-pair on $\mathcal{T}$. When we sample a tree $\mathcal{T}$, by Property 4 (Lemma 16), the total number of tree-demand-pairs in $\mathcal{T}$ is $\psi \leqslant qn^{2\ell} \cdot 2^{2\ell\beta}$. Set

$$\tau_2 = \frac{1}{\phi} \cdot \log(4qn^{2\ell} \cdot 2^{2\ell\beta}) = O(\log q + \ell(\log n + \beta)).$$

We can connect every tree-demand-pairs on $\mathcal{T}$ (denote this event by $\mathcal{C}_\mathcal{T}$) with probability:

$$\Pr[\mathcal{C}_\mathcal{T}] \geqslant 1 - \psi \cdot (1-\phi)^{\tau_2} \geqslant 3/4.$$

Conditioned on $\mathcal{C}_\mathcal{T}$ for certain $\mathcal{T}$ that is good for F, it holds that $H_{\ell+1} \backslash F$ is connected for all (S_i, T_i) by Property 2 (Lemma 11). Next, because for every F, by Lemma 8, we have $1/2$ probability such that $\mathcal{T}$ is good for F. Set

$$\tau_1 = \frac{8}{3} \log 2n^{2\ell} = O(\ell \log n).$$

Let $\psi' \leqslant n^{2\ell}$ be the number of possible edge-cut F. The probability that we sample a good tree $\mathcal{T}$ for all F and $\mathcal{T}$ is fully connected ($\mathcal{C}_\mathcal{T}$) is at least:

$$1 - \psi' \cdot (1 - 3/4 \cdot 1/2)^{\tau_1} \geqslant 1/2.$$

It means that $H_{\ell+1} \backslash F$ is connected for every (S_i, T_i) and for all F with probability at least $1/2$, which concludes the lemma. □

Lemma 10 (Cost). *The output graph $H_{\ell+1}$ of Algorithm 1 has cost*

$$O(\ell^2 \beta^2 \log^4 n(\ell \log n + \ell\beta + \log q)) \sum_{e \in E} c_e x_e.$$

Proof. We use $4\ell\beta \cdot \sum_{e\in E} x_e = O(\ell\beta) \sum_{e \in E} x_e$ cost to buy the edges in LARGE. For the multiple rounds of TreeRounding, by Lemma 7 and Property 3 (Lemma 14), we know that for each round we pay

$$O(\ell\beta^2 \log^3 n) \cdot \sum_{e\in E} c_e x_e.$$

As $\tau_1 = O(\ell \log n)$, $\tau_2 = O(\log q + \ell(\log n + \beta))$, the total cost incurred by the algorithm is

$$O(\ell \log n \cdot (\log q + \ell(\log n + \beta)) \cdot \ell\beta^2 \cdot \log^3 n) \sum_{e \in E} c(e) x_e,$$

as we claimed. □

Conclusion Combining Lemma 9 and Lemma 10, Algorithm 1 is a Monte Carlo $O(\ell^2\beta^2\log^4 n(\ell\log n + \ell\beta + \log q))$-approximation algorithm. It can be viewed as a Las Vegas one if we keep sampling until $H_{\ell+1}$ is $(\ell+1)$-connected. It concludes the approximation ratio of

$$O(\beta^2 k^2 \log^4 n(k\log n + k\beta + \log q))$$

as Theorem 2 claims. By using Räcke's tree embedding where $\beta = O(\log n)$, we have the approximation ratio of

$$O(k^2\log^6 n(k\log n + \log q))$$

as that in Theorem 1.

5.1 Tree-Demand-Pairs for Shattered-Component-Level Paths

Let $\mathbb{Q}^F$ be the set of all connected components in $H\backslash F$. We remark that the subgraph H might already consist of many connected components before removing the edges in F, for $V(H) = V(G)$ but $|E(H)|$ might be much smaller than $|E(G)|$. For convenience, we think of each component $Q \in \mathbb{Q}^F$ as a subset of V instead of a subgraph of H. For a nonempty set $S \subseteq V(G)$, suppose that $Q_1, Q_2, \ldots, Q_t$ are the components in $\mathbb{Q}^F$ such that $Q_i \bigcap S \neq \varnothing$ for $1 \leqslant i \leqslant t$, we define $Q_S = \bigcup_{i=1}^t Q_i$.

For a cut $X \subseteq V(G)$, we say that X is *shattered* if there exists some component $Q \in \mathbb{Q}^F$ such that $X \bigcap Q \neq \varnothing$ and $(V(G)\backslash X) \bigcap Q \neq \varnothing$; otherwise, X is *intact*. For a tree $\mathcal{T} \in \mathcal{D}$ and a component $Q \in \mathbb{Q}^F$, we say that Q is *shattered* if it is disconnected in $\mathcal{T}\backslash\mathcal{M}^{-1}(F)$; otherwise, Q is *intact*.

Now we are ready to generate a collection Z_F of tree-demand-pairs for each edge-set F of ℓ edges and a good tree $\mathcal{T} \in \mathcal{D}$ for F. We generate all the possible partitions (A', B') of shattered components in $\mathbb{Q}^F$, denoted by $\mathbb{U}^F_{\mathcal{T}}$. Here we allow $A' = \varnothing$ or $B' = \varnothing$. For each partition (A', B'), generate tree-demand-pairs $(A' \bigcup Q_{S_i}, B' \bigcup Q_{T_i})$ for each $i \in [q]$ such that $Q_{S_i} \bigcap Q_{T_i} = \varnothing$. Formally, we have

$$\begin{aligned} Z_F := \{(A' \bigcup Q_{S_i}, B' \bigcup Q_{T_i}) \mid (A', B') \in \mathbb{U}^F_{\mathcal{T}}, \\ Q_{S_i} \bigcap Q_{T_i} = \varnothing, i \in [q]\} \end{aligned}$$

Notice that we only need Z_F as tree-demand-pairs on tree $\mathcal{T}$ when $\mathcal{T}$ is *good* for F, i.e., F is *good* on $\mathcal{T}$. The set of all tree-demand-pairs for a tree $\mathcal{T}$ is denoted by $Z_{\mathcal{T}} = \bigcup_{F\subseteq E(H):|F|=\ell, F \text{ is good on } \mathcal{T}} Z_F$.

5.2 Feasibility of the Reduction

We prove that the set of tree-demand-pairs of Z_F on a tree $\mathcal{T}$ corresponds to the problem of connecting all set pairs (S_i, T_i) that are disconnected in $H\backslash F$.

Lemma 11. *For an edge-set $F \subseteq E(H)$ of ℓ edges, if there exist a tree $\mathcal{T}$ and a set of edges $E_b \subseteq E(\mathcal{T})\backslash\mathcal{M}^{-1}(F)$ such that for every demand pair $(A, B) \in Z_F$ there is a path in E_b connecting A to B, then there is a (S_i, T_i)-path in $(H \bigcup \mathcal{M}(E_b))\backslash F$ for each $i \in [q]$.*

Proof. Let $\overline{H} = H \bigcup \mathcal{M}(E_b)$ and $V = V(G) = V(H)$. We recall that, for a vertex set $S \subseteq V$, Q_S is defined as the union of components in $\mathbb{Q}^F$ intersecting with S, where $\mathbb{Q}^F$ consists of all the connected components in $H\backslash F$ (a component is viewed as a subset of V rather than a subgraph of H). If $Q_{S_i} \bigcap Q_{T_i} \neq \varnothing$, then we are done. Hence, we may assume that $Q_{S_i} \bigcap Q_{T_i} = \varnothing$. Then it suffices to show that, $\delta_{\overline{H}}(X) \geqslant 1$, for every cut $X \subseteq V$ satisfying $Q_{S_i} \subseteq X$ and $Q_{T_i} \subseteq V \backslash X$.

If X is a shattered cut, then there is some component $Q \in \mathbb{Q}^F$ such that $Q \bigcap X \neq \varnothing$ and $Q \bigcap (V \backslash X) \neq \varnothing$. Thus, there is an edge $e = (u, v) \in E(H) \backslash F \subseteq E(\overline{H})$ such that $u \in Q \bigcap X$ and $v \in Q \bigcap (V \backslash X)$, implying that $\delta_{\overline{H}}(X) \geqslant 1$.

If X is an intact cut, then let $A = Q_{S_i} \bigcup \{Q \in \mathbb{Q}^F : Q \subseteq X\}$ and $B = Q_{T_i} \bigcup \{Q \in \mathbb{Q}^F : Q \subseteq V \backslash X\}$. Clearly, it holds that $(A, B) \in Z_F$. By the premise, there is a path p in E_b connecting $A \subseteq X$ to $B \subseteq V \backslash X$. Thus, it follows from the properties of Räcke's tree that the corresponding path $\mathcal{M}(p)$ is in $H \bigcup \mathcal{M}(E_b)$. The premise $E_b \subseteq E(\mathcal{T}) \backslash \mathcal{M}^{-1}(F)$ further ensures that $\mathcal{M}(p)$ is in $(H \bigcup \mathcal{M}(E_b)) \backslash F$ because, otherwise, p would use some edge in $\mathcal{M}^{-1}(F)$. Thus, $\delta_{\overline{H}}(X) \geqslant 1$, proving the lemma. □

5.3 Bounding the Flow and the Number of Tree-Demand-Pairs

As mentioned, we will later invoke a randomized algorithm to round the flows on all sampled trees $\mathcal{T}$. However, given a specific tree $\mathcal{T}$, it might be infeasible to connect the tree-demand-pairs for all the edge-sets. This is because the maximum flow of some tree-demand-pairs may be too small. To fix this problem, first let us recall the definition of "good".

Definition 12. *We say that a tree $(\mathcal{T}, M, y)$ is good for an edge-set $F \subseteq E(H)$, or F is good on $\mathcal{T}$, if $y(\mathcal{M}^{-1}(F)) \leqslant 1/2$.*

We recall that there are two types of cuts $X \subseteq V(H)$: intact and shattered. We first prove that an intact cut has at least constant capacity with respect to $\widetilde{x}$ in the graph G.

Lemma 13. *Let $X \subseteq V(G)$ be a cut such that $S_i \subseteq X$ and $T_i \subseteq V \backslash X$ for some set-pair (S_i, T_i). If X is intact with respect to F, then $\widetilde{x}(\delta_G(X) \backslash F) \geqslant 3/4$.*

Proof. Since X is intact, all edges in $\delta_G(X) \backslash F$ are not in H, meaning that $\delta_G(X) \backslash F \subseteq \mathsf{SMALL}$. Thus, each edge e in $\delta_G(X) \backslash F$ has capacity $\widetilde{x}_e = x_e$. It follows from the LP constraints on x that $x(\delta_G(X)) \geqslant \ell + 1$. Since F consists of ℓ edges, the capacity $x(F)$ is at most ℓ. Thus,

$$\ell + 1 \leqslant x(\delta_G(X)) = x(\delta_G(X) \backslash F) + x(F) \leqslant x(\delta_G(X) \backslash F) + \ell$$

Therefore, $x(\delta_G(X) \backslash F) \geqslant 1$, implying that

$$\begin{aligned} \widetilde{x}(\delta_G(X) \backslash F) &= x(\delta_G(X) \backslash F) - \sum_{e \in \delta_G(X) \backslash F :\, x_e < (1/2n^2) \cdot (1/4\ell\beta)} x_e \\ &\geqslant 1 - n^2 \cdot (1/2n^2) \cdot (1/4\ell\beta) \\ &\geqslant 3/4. \end{aligned}$$

□

Since the capacity of an edge-set F in any good tree is at most $2\beta \cdot \widetilde{x}(F)$, the next lemma follows.

Lemma 14 (Lower Bound on the Flow Value in the Tree)**.** *Let $F \subseteq E(H)$ be any edge-set of ℓ edges such that $Q_{S_i} \bigcap Q_{T_i} = \varnothing$ for some $i \in [q]$ and let (A, B) be any tree-demand-pair generated for S_i and T_i. Then for a good tree $\mathcal{T}$ for F the flow that can be routed from A to B in $\mathcal{T} \backslash \mathcal{M}^{-1}(F)$ is at least $1/(4\ell\beta)$.*

Proof. It is sufficient to prove that, for any $X \subseteq V$ such that $A \subseteq X$ and $B \subseteq V \backslash X$, the value of flow that can be routed in $\mathcal{T} \backslash \mathcal{M}^{-1}(F)$ from X to $V \backslash X$ is at least $1/(4\ell\beta)$.

If X is an intact cut, then $\widetilde{x}(\delta_G(X) \backslash F) \geqslant 3/4$ by Lemma 13 and $\mathsf{flow}^y_{\mathcal{T}}(X, V \backslash X) \geqslant \mathsf{flow}^{\widetilde{x}}_G(X, V \backslash X) \geqslant 3/4$. Since F is good on $\mathcal{T}$, we have $y(\mathcal{M}^{-1}(F)) \leqslant 1/2$ which means that if the edges $\mathcal{M}^{-1}(F)$ are removed from $\mathcal{T}$, then any flow in $\mathcal{T}$ can decrease in value by at most $1/2$. Therefore, we have that

$$\begin{aligned}\mathsf{flow}^y_{\mathcal{T} \backslash \mathcal{M}^{-1}(F)}(X, V \backslash X) &\geqslant \mathsf{flow}^y_{\mathcal{T}}(X, V \backslash X) - 1/2 \\ &\geqslant 1/4 \geqslant 1/(4\ell\beta)\end{aligned}$$

If X is a shattered cut, then there is some component $Q \in \mathbb{Q}^F$ such that $Q \bigcap X \neq \emptyset$ and $Q \bigcap (V \backslash X) \neq \emptyset$. The definition of tree-demand-pair (A, B) implies that Q cannot be a shattered component. Thus, Q is still connected in $\mathcal{T} \backslash \mathcal{M}^{-1}(F)$, and there is a path in $\mathcal{T} \backslash \mathcal{M}^{-1}(F)$ connecting X to $V \backslash X$.

Consider an arbitrary pair s, t such that $s \in Q \bigcap X$ and $t \in Q \bigcap (V \backslash X)$. We know that s and t are connected by LARGE edges in H, so $\mathsf{flow}^{\widetilde{x}}_G(s, t) \geqslant \frac{1}{4\ell\beta}$. Notice that on the tree, s and t are still connected in $T \backslash \mathcal{M}^{-1}(F)$. Therefore, the unique tree path for (s, t) does not contain F and we have:

$$\begin{aligned}\mathsf{flow}^y_{\mathcal{T} \backslash \mathcal{M}^{-1}(F)}(X, V \backslash X) &\geqslant \mathsf{flow}^y_{\mathcal{T} \backslash \mathcal{M}^{-1}(F)}(s, t) \\ &= \mathsf{flow}^y_{\mathcal{T}}(s, t) \geqslant \mathsf{flow}^{\widetilde{x}}_G(s, t) \geqslant \frac{1}{4\ell\beta}\end{aligned}$$ □

Lemma 14 implies that for every edge-set $F \subseteq E(H)$ there is enough flow in a good tree to connect each tree-demand-pair $(A, B) \in Z_F$. Next, we bound the number of tree-demand-pairs.

Lemma 15. *For any edge-set $F \subseteq E(H)$ of ℓ edges and a good tree $\mathcal{T}$ for F, the number of shattered components in $\mathbb{Q}^F$ is at most $2\ell\beta$.*

Proof. Let $Q \in \mathbb{Q}^F$ be any shattered component. By definition, there exist two vertices $s, t \in Q$ that are disconnected in $\mathcal{T} \backslash \mathcal{M}^{-1}(F)$, while being connected in $H \backslash F$. Since each edge in $H \backslash F$ has capacity $1/(4\ell\beta)$, the maximum flow from s_i to t_i in $H \backslash F$ has value at least $1/(4\ell\beta)$ and so do these two vertices in the tree T. However, s and t are disconnected in $\mathcal{T} \backslash \mathcal{M}^{-1}(F)$, meaning that there cannot be any flow between them in $\mathcal{T} \backslash \mathcal{M}^{-1}(F)$. Applying this fact to every shattered component, we conclude that the total loss of flow value is at least $\nu/(4\ell\beta)$ after removing edges $\mathcal{M}^{-1}(F)$ from $\mathcal{T}$, where ν is the number of shattered components. Therefore

$$\nu/(4\ell\beta) \leqslant y(\mathcal{M}^{-1}(F)) \leqslant 1/2.$$

Hence we conclude that $\nu \leqslant 2\ell\beta$. □

Lemma 16 (Upper Bound on the Number of Tree-Demand-Pairs)**.** *For each tree $\mathcal{T}$, the number of tree-demand-pairs is $|Z_{\mathcal{T}}| \leqslant q \cdot n^{2\ell} \cdot 2^{2\ell\beta}$.*

Proof. By Lemma 15 we have

$$\begin{aligned}|Z_{\mathcal{T}}| &\leqslant \sum_{F \subseteq E(H) \colon |F| = \ell, F \text{ is good on } \mathcal{T}} |Z_F| \\ &\leqslant \binom{|E(H)|}{\ell} \cdot (2^{2\ell\beta} \cdot q) \\ &\leqslant q \cdot n^{2\ell} \cdot 2^{2\ell\beta}.\end{aligned}$$ □

6 Rounding Flows on the Tree

For completeness, we illustrate how we apply the CGL's rounding in this section. The algorithm is shown in Algorithm 2.

Algorithm 2 TreeRounding

```
for $q = 0, ..., 2\log(\frac{2n^2}{f})$ and $v \in V(\mathcal{T})$ do
    With probability $\min\{1, \frac{8}{f} \cdot 2^{-q}\}$, $q$-mark $v$.
    if $v$ is $q$-marked then
        $y^{v,q} \leftarrow 2^{q+1}y$ and $E^{v,q} \leftarrow \emptyset$.
        for $\tau'$ interations do
            $E^{v,q} \leftarrow E^{v,q} \cup \text{RoundGKR}(\mathcal{T}_v, v, y^{v,q})$.
        $H \leftarrow H \cup \mathcal{M}(E^{v,q})$.
return $H$.
```

We prove the following Lemma 7 by combining Lemma 18 and Lemma 19.

Lemma 7 (CGL's Tree Rounding)**.** *Suppose that there is a tree $\mathcal{T}$ with $\mathsf{height}(\mathcal{T}) = O(\log n)$ and an edge set $E' \subseteq E(\mathcal{T})$ that supports a flow of value at least f between two vertex sets A and B. If only edges from E' can be selected by TreeRounding$(G, \mathcal{T})$, then it connects A to B with constant probability ϕ with cost of $O(\frac{1}{f} \cdot \beta \cdot \log^3 n) \cdot \sum_{e \in E} c_e x_e$.*

First, we introduce Garg-Konjevod-Ravi (GKR) Rounding which is repeatedly used in the TreeRounding algorithm. Denote the rounding algorithm for GST used in [GKR00] by RoundGKR. Let $\mathcal{T}$ be the tree rooted at r, and y be a fractional solution to the standard cut-based LP for GST. RoundGKR gives a way to connect every group $\mathcal{G}_i$ to root r with high probability. To be more precise, we state the following result implicitly shown by the authors.

Lemma 17. *Suppose that, for some $E' \subseteq E(\mathcal{T})$, capacity y support an unit flow from $\mathcal{G}_i \subseteq V(\mathcal{T})$ to r in E'. RoundGKR$(\mathcal{T}, r, y)$ where $\mathcal{T}$ is a tree using fractional solution y and r is the root of $\mathcal{T}$, gives a path $P \subseteq E'$ connects $\mathcal{G}_i$ to r with probability at least $\Omega(1/\log n)$.*

In the TreeRounding algorithm, we have a tree embedding $\mathcal{T}$ initially. For one iteration q and node v, with some probability w.r.t q, it applies the subroutine RoundGKR for τ' iterations. In the end, the union of the corresponding edges in the original graph of the edges selected by the RoundGKR gives a solution to connect A and B.

Now we are ready to prove Lemma 7. First, we show the probability that the algorithm connects each demand-pair A and B.

Lemma 18. *Algorithm 2 connects A to B with constant probability ϕ.*

Proof. We know that there is an f unit of flow from A to B in E'. Then, we decompose this flow into a family of flow paths. By discarding all the flow paths whose flow is less than $\frac{1}{2n^2} \cdot f$, we obtain a new family of flow path P. Denote the flow of P_j by p_j. Let μ_v be the total amount of flow turning at v. Then we have

$$\sum_{v \in V(\mathcal{T})} \mu_v = \sum_{j=1}^{|P|} p_j \geqslant f - \frac{1}{2n^2} \cdot f \cdot |E(T)| \geqslant \frac{f}{2}.$$

For each node $v \in V(\mathcal{T})$, let $q_v \in \left\{0, ..., 2\log(\frac{2n^2}{f})\right\}$ be an integer such that $\mu_v \in (2^{-q_v-1}, 2^{-q_v}]$.

Denote the event that some node $v \in V(\mathcal{T})$ is q_v-marked by $\mathcal{I}$. We claim that $\mathbf{Pr}[\bar{\mathcal{I}}|\mathcal{G}] \leqslant e^{-3/4}$. Let X_v be 1 if node $v \in V(\mathcal{T})$ is q_v-marked and 0 otherwise. We have

$$\mathbf{Pr}\left[\bar{\mathcal{I}}|\mathcal{G}\right] = \mathbf{Pr}\left[\sum_{v \in V(T)} X_v = 0\right].$$

Since $\{X_v\}$ are independent,

$$\mathbb{E}\left[\sum_{v \in V(\mathcal{T})} X_v\right] = \sum_{v \in V(\mathcal{T})} \frac{8}{f} \cdot 2^{-q_v} \geqslant \frac{8}{f} \sum_{v \in V(\mathcal{T})} p_v \geqslant 4.$$

By Chernoff's bound,

$$\mathbf{Pr}\left[\sum_{v \in V(\mathcal{T})} X_v \leqslant 1\right] \leqslant e^{-\frac{1}{3}(\frac{3}{4})^2 4} = e^{-3/4}.$$

Conditioned on $\mathcal{I}$, there is some node $v \in V(\mathcal{T})$ that is q_v-marked. Consider the event $\mathcal{C}$ that A and B are connected by the union of the solution computed by RoundGKR in τ' iterations on node v for $q = q_v$. Observe that $2^{q_v+1}\phi_v \geqslant 1$. Therefore, by Lemma 17, RoundGKR selects a correct path with probability at least $1 - (1 - \Omega(\frac{1}{\log n}))^{\tau'}$. Thus, $\mathbf{Pr}[\bar{\mathcal{C}}|\mathcal{G}, \mathcal{I}] \leqslant \varepsilon$, where $\tau' = O(\log n)$. Altogether, the probability that the Tree Rounding connects A to B is at least

$$\begin{aligned} 1 - \mathbf{Pr}[\bar{\mathcal{G}}] - \mathbf{Pr}[\bar{\mathcal{I}}|\mathcal{G}] - \mathbf{Pr}[\mathcal{G}] \cdot \mathbf{Pr}[\mathcal{I}|\mathcal{G}] \cdot \mathbf{Pr}[\bar{\mathcal{C}}|\mathcal{G}, \mathcal{I}] \\ \geqslant 1 - 1/2 - e^{-3/4} - \varepsilon =: \phi > 0 \end{aligned}$$ □

Next we analyze the cost incurred by the algorithm.

Lemma 19. $\mathbb{E}[c(H)] = O(\frac{1}{f} \cdot \beta \cdot \log^3 n) \sum_{e \in E} c_e x_e$.

Proof. First, an edge $e' \in E(\mathcal{T})$ is selected iff e' is in $E^{v,q}$ for some q and some q-marked $v \in V(\mathcal{T})$. For each iteration of RoundGKR, we select e' with probability $y^{v,q} = 2^{q+1}y$. To proceed these iterations, the vertex v need to be q-marked. Thus, the corresponding probability is at most $\frac{8}{f} \cdot 2^{-q}$. Putting everything together, edge e is selected with probability at most

$$\mathsf{height}(\mathcal{T}) \cdot \sum_q \frac{8}{f} \cdot 2^{-q} \cdot 2^{q+1} y(e') \cdot \tau' = O\left(\frac{\log^3 n}{f}\right) y(e').$$

Then, an edge $e \in E$ is selected iff any edge in $\mathcal{M}^{-1}(e)$ is selected by RoundGKR. Therefore, the expectation of an edge e is selected is at most

$$O\left(\frac{\log^3 n}{f}\right) \cdot \mathbb{E}\left[\sum_{e' \in \mathcal{M}^{-1}(e)} y(e')\right] \leqslant O\left(\frac{1}{f} \cdot \beta \cdot \log^3 n\right) \cdot x_e.$$

Taking the summation of all edges, the expected cost of $c(H)$ is at most

$$O\left(\frac{1}{f} \cdot \beta \cdot \log^3 n\right) \sum_{e \in E} c(e) x_e.$$ □

7 Conclusion and Open Problems

In this paper, we have presented an approximation algorithm for Group EC-SNDP whose approximation ratio is $O(k^3 \operatorname{polylog}(n, q))$. It is quite interesting that this factor resemblances the approximation ratio of $O(k^3 \log n)$ for VC-SNDP [CK12]. The factor k appears quite naturally in network design problems and might be the approximability threshold. However, the best known negative result still has a lower bound of $k^{1/5-\epsilon}$, for $\epsilon > 0$, assuming $\mathsf{NP} \neq \mathsf{ZPP}$, [CDE+18] (combined with [Lae14] and the improvement in [Man19]). While it is rather unnatural that a survivable network design problem would have a hardness factor beyond k, the recent result of Liao, Chen, Laekhanukit and Zhang [LCLZ22] shows that this is the case for the sister problem of Group EC-SNDP, namely the *k-connected directed Steiner tree* problem (k-DST) (a.k.a, *directed single-source k-connectivity*). They showed that the approximation lower bound of k-DST is, indeed, at least $\Omega(2^{k/2}/k)$, which is far beyond polynomial. Any results in both directions would be surprisingly interesting, either the existence of $k^{1-\epsilon}\operatorname{polylog}(n)$-approximation algorithms or a hardness threshold of $O(k^{1+\epsilon})$, for some $\epsilon > 0$.

We are aware of the capacity-based probabilistic tree embedding for *α-balanced* graphs, which appears in the work of Ene, Miller, Pachocki and Sidford [EMPS16]. In our opinion, there is a high chance that the probabilistic capacity mapping would work for us, thus generalizing our framework to directed graphs, or more specifically, to the case of k-DST on α-balanced graphs. Unfortunately, the construction in [EMPS16] is tailored for minimizing congestion for single-source oblivious routing, and it is not clear whether the bound holds for other set-pairs in the graph. Our algorithm, on the other hand, requires the congestion guarantee to hold for pairwise subsets, in which one contains the source (i.e., root). Thus, one direction to push forward on studying survivable network design is in developing a capacity-based probabilistic tree-embedding that is able to deal with the more general settings of oblivious routing on α-balanced graphs.

References

[AF09] Reid Andersen and Uriel Feige. Interchanging distance and capacity in probabilistic mappings. *CoRR*, abs/0907.3631, 2009.

[BCE+20] Glencora Borradaile, Erin Wolf Chambers, David Eppstein, William Maxwell, and Amir Nayyeri. Low-stretch spanning trees of graphs with bounded width. In Susanne Albers, editor, *17th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2020, June 22-24, 2020, Tórshavn, Faroe Islands*, volume 162 of *LIPIcs*, pages 15:1–15:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[CCK08] Tanmoy Chakraborty, Julia Chuzhoy, and Sanjeev Khanna. Network design for vertex connectivity. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 167–176. ACM, 2008.

[CDE+18] Parinya Chalermsook, Syamantak Das, Guy Even, Bundit Laekhanukit, and Daniel Vaz. Survivable network design for group connectivity in low-treewidth graphs. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, AP-*

PROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA, volume 116 of *LIPIcs*, pages 8:1–8:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[CGL15] Parinya Chalermsook, Fabrizio Grandoni, and Bundit Laekhanukit. On survivable set connectivity. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 25–36. SIAM, 2015.

[CK12] Julia Chuzhoy and Sanjeev Khanna. An o(k^{3}log n)-approximation algorithm for vertex-connectivity survivable network design. *Theory Comput.*, 8(1):401–413, 2012.

[CLNV14] Joseph Cheriyan, Bundit Laekhanukit, Guyslain Naves, and Adrian Vetta. Approximating rooted steiner networks. *ACM Transactions on Algorithms (TALG)*, 11(2):1–22, 2014.

[CLWZ20] Chun-Hsiang Chan, Bundit Laekhanukit, Hao-Ting Wei, and Yuhao Zhang. Polylogarithmic approximation algorithm for k-connected directed steiner tree on quasi-bipartite graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176, pages 63:1–63:20. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[CV14] Joseph Cheriyan and László A. Végh. Approximating minimum-cost k-node connected subgraphs via independence-free graphs. *SIAM J. Comput.*, 43(4):1342–1362, 2014.

[Din16] Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electron. Colloquium Comput. Complex.*, page 128, 2016.

[DK99] Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, page 750–759, New York, NY, USA, 1999. Association for Computing Machinery.

[DM18] Irit Dinur and Pasin Manurangsi. Eth-hardness of approximating 2-csps and directed steiner network. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 36:1–36:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[Eme11] Yuval Emek. *K*-outerplanar graphs, planar duality, and low stretch spanning trees. *Algorithmica*, 61(1):141–160, sep 2011.

[EMPS16] Alina Ene, Gary Miller, Jakub Pachocki, and Aaron Sidford. Routing under balance. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 598–611, New York, NY, USA, 2016. Association for Computing Machinery.

[FJW06] Lisa Fleischer, Kamal Jain, and David P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *J. Comput. Syst. Sci.*, 72(5):838–867, aug 2006.

[FL12] Jittat Fakcharoenphol and Bundit Laekhanukit. An o($\log^2 k$)-approximation algorithm for the k-vertex connected spanning subgraph problem. *SIAM J. Comput.*, 41(5):1095–1109, 2012.

[FNR15] Takuro Fukunaga, Zeev Nutov, and R. Ravi. Iterative rounding approximation algorithms for degree-bounded node-connectivity network design. *SIAM J. Comput.*, 44(5):1202–1229, 2015.

[GGP$^+$94] M. X. Goemans, A. V. Goldberg, S. 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*, SODA '94, page 223–232, USA, 1994. Society for Industrial and Applied Mathematics.

[GKR00] Naveen Garg, Goran Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000.

[GKR10] Anupam Gupta, Ravishankar Krishnaswamy, and R. Ravi. Tree embeddings for two-edge-connected network design. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1521–1538. SIAM, 2010.

[GL17] Fabrizio Grandoni and Bundit Laekhanukit. Surviving in directed graphs: a quasi-polynomial-time polylogarithmic approximation for two-connected directed steiner tree. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 420–428, 2017.

[Jai01] Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Comb.*, 21(1):39–60, 2001.

[KKL04] Guy Kortsarz, Robert Krauthgamer, and James R. Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM J. Comput.*, 33(3):704–720, 2004.

[KKN12] Rohit Khandekar, Guy Kortsarz, and Zeev Nutov. Approximating fault-tolerant group-steiner problems. *Theor. Comput. Sci.*, 416:55–64, 2012.

[KM05] Hervé Kerivin and A. Ridha Mahjoub. Design of survivable networks: A survey. *Netw.*, 46(1):1–21, aug 2005.

[Lae14] Bundit Laekhanukit. Parameters of two-prover-one-round game and the hardness of connectivity problems. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1626–1643. SIAM, 2014.

[LCLZ22] Chao Liao, Qingyun Chen, Bundit Laekhanukit, and Yuhao Zhang. Almost tight approximation hardness for single-source directed k-edge-connectivity. 229:89:1–89:17, 2022.

[Man19] Pasin Manurangsi. A note on degree vs gap of min-rep label cover and improved inapproximability for connectivity problems. *Information Processing Letters*, 145:24–29, 2019.

[MRS21] Pasin Manurangsi, Aviad Rubinstein, and Tselil Schramm. The strongish planted clique hypothesis and its consequences. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 10:1–10:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[Nut12] Zeev Nutov. Approximating minimum-cost connectivity problems via uncrossable bifamilies. *ACM Transactions on Algorithms (TALG)*, 9(1):1–16, 2012.

[Nut18] Zeev Nutov. Node-connectivity survivable network problems. In Teofilo F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 2: Contemporary and Emerging Applications*. Chapman and Hall/CRC, 2018.

[Nut20] Zeev Nutov. A $4 + \epsilon$ approximation for k-connected subgraphs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1000–1009. SIAM, 2020.

[Nut21] Zeev Nutov. On rooted k-connectivity problems in quasi-bipartite digraphs. In Rahul Santhanam and Daniil Musatov, editors, *Computer Science - Theory and Applications - 16th International Computer Science Symposium in Russia, CSR 2021, Sochi, Russia, June 28 - July 2, 2021, Proceedings*, volume 12730 of *Lecture Notes in Computer Science*, pages 339–348. Springer, 2021.

[Räc08] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 255–264, 2008.

[SWK69] K. Steiglitz, P. Weiner, and D. Kleitman. The design of minimum-cost survivable networks. *IEEE Transactions on Circuit Theory*, 16(4):455–460, 1969.

[WGMV93] David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani. A primal-dual approximation algorithm for generalized steiner network problems. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, page 708–717, New York, NY, USA, 1993. Association for Computing Machinery.