Approximation Algorithms for Network Design in Non-Uniform Fault Models

Chandra Chekuri
Department of Computer Science, University of Illinois, Urbana-Champaign, Urbana, IL, USA

Rhea Jain
Department of Computer Science, University of Illinois, Urbana-Champaign, Urbana, IL, USA

Abstract

Classical network design models, such as the Survivable Network Design problem (SNDP), are (partly) motivated by robustness to faults under the assumption that any subset of edges upto a specific number can fail. We consider *non-uniform* fault models where the subset of edges that fail can be specified in different ways. Our primary interest is in the flexible graph connectivity model [1, 3, 4, 8], in which the edge set is partitioned into *safe* and *unsafe* edges. Given parameters $p, q \geq 1$, the goal is to find a cheap subgraph that remains p-connected even after the failure of q unsafe edges. We also discuss the bulk-robust model [6, 2] and the relative survivable network design model [19]. While SNDP admits a 2-approximation [32], the approximability of problems in these more complex models is much less understood even in special cases. We make two contributions.

Our first set of results are in the flexible graph connectivity model. Motivated by a conjecture that a constant factor approximation is feasible when p and q are fixed, we consider two special cases. For the s-t case we obtain an approximation ratio that depends only on p, q whenever $p + q > pq/2$ which includes $(p, 2)$ and $(2, q)$ for all $p, q \geq 1$. For the global connectivity case we obtain an $O(q)$ approximation for $(2, q)$, and an $O(p)$ approximation for $(p, 2)$ and $(p, 3)$ for any $p \geq 1$, and for $(p, 4)$ when p is even. These are based on an augmentation framework and decomposing the families of cuts that need to be covered into a small number of uncrossable families.

Our second result is a poly-logarithmic approximation for a generalization of the bulk-robust model when the "width" of the given instance (the maximum number of edges that can fail in any particular scenario) is fixed. Via this, we derive corresponding approximations for the flexible graph connectivity model and the relative survivable network design model. We utilize a recent framework due to Chen et al. [17] that was designed for handling group connectivity.

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

Keywords and phrases non-uniform faults, network design, approximation algorithm

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.36

Category Track A: Algorithms, Complexity and Games

Funding *Chandra Chekuri*: Supported in part by NSF grants CCF-1910149 and CCF-1907937.
Rhea Jain: Supported in part by NSF grant CCF-1907937.

Acknowledgements We thank Qingyun Chen for clarifications on a proof in [17]. We thank Joseph Cheriyan for pointers and helpful comments on flexible graph connectivity. The initial impetus for our work on this topic came from [8].

1 Introduction

The Survivable Network Design Problem (SNDP) is an important problem in combinatorial optimization that generalizes many well-known problems related to connectivity and is also motivated by practical problems related to the design of fault-tolerant networks. The input to this problem is an undirected graph $G = (V, E)$ with non-negative edge costs $c : E \to \mathbb{R}_+$ and a collection of source-sink pairs $(s_1, t_1), \ldots, (s_h, t_h)$, each with an integer connectivity requirement r_i. The goal is to find a minimum-cost subgraph H of G such that

H has r_i connectivity for each pair (s_i, t_i). We focus on edge-connectivity requirements in this paper.[1] SNDP contains as special cases classical problems such as s-t shortest path, minimum spanning tree (MST), minimum k-edge-connected subgraph (k-ECSS), Steiner tree, Steiner forest and several others. It is NP-Hard and APX-Hard to approximate. There is a 2-approximation via the iterated rounding technique [32].

A pair (s, t) that is k-edge-connected in G is robust to the failure of *any* set of $k - 1$ edges. In various settings, the set of edges that can fail can be correlated and/or exhibit non-uniform aspects. We are interested in network design in such settings, and discuss a few models of interest that have been studied in the (recent) past. We start with the flexible graph connectivity model (flex-connectivity for short) that was the impetus for our work.

Flexible graph connectivity. In this model, first introduced by Adjiashvili [1] and studied in several recent papers [3, 4, 5, 8, 7], the input is an edge-weighted undirected graph $G = (V, E)$ where the edge set E is partitioned to *safe* edges $\mathcal{S}$ and *unsafe* edges $\mathcal{U}$. The assumption, as the names suggest, is that unsafe edges can fail while safe edges cannot. We say that a vertex-pair (s, t) is (p, q)-flex-connected in a subgraph H of G if s and t are p-edge-connected after deleting from H any subset of at most q unsafe edges. The input, as in SNDP, consists of G and h source-sink pairs; the i'th pair now specifies a (p_i, q_i)-flex-connectivity requirement. The goal is to find a min-cost subgraph H of G such that for each i, s_i and t_i are (p_i, q_i)-flex-connected in H. We refer to this as the Flex-SNDP problem. Note that Flex-SNDP generalizes SNDP in two ways[2].

Bulk-robust network design. This fairly general non-uniform model was introduced by Adjiashvili, Stiller and Zenklusen [6]. Here an explicit *scenario* set $\Omega = \{F_1, F_2, \ldots, F_m\}$ is given as part of the input where each $F_j \subseteq E$. The goal is to find a min-cost subgraph H of G such that each of the given pairs (s_i, t_i) remains connected in $H - F_j$ for each $j \in [m]$. We consider a slight generalization of this problem in which each scenario is now a pair $(F_j, \mathcal{K}_j)$ where $\mathcal{K}_j$ is a set of source-sink pairs. As earlier, the goal is to find a min-cost subgraph H of G such that for each $j \in [m]$, each pair (s_i, t_i) in $\mathcal{K}_j$ is connected in $H - F_j$. The *width* of the failure scenarios is $\max_{1 \le j \le \ell} |F_j|$. We use Bulk-SNDP to refer to this problem.

The advantage of the bulk-robust model is that one can specify arbitrarily correlated failure patterns, allowing it to capture many well studied problems in network design. We observe that SNDP and Flex-SNDP problem can be cast as special cases of Bulk-SNDP model where the width is $\max_i(r_i - 1)$ in the former case, and $\max_i(p_i + q_i - 1)$ in the latter case. The slight generalization on Bulk-SNDP described above also allows us to model a new problem recently proposed by Dinitz, Koranteng, and Kortsarz [19] called **Relative Survivable Network Design** (RSNDP). This problem allows one to ask for higher connectivity even when the underlying graph G has small cuts. The input is an edge-weighted graph $G = (V, E)$ and source-sink pairs (s_i, t_i) each with requirement r_i; the goal is to find a min-cost subgraph H of G such that for each $F \subseteq E$ with $|F| < r_i$, (s_i, t_i) is connected in $H - F$ if s_i and t_i are connected in $G - F$. It is easy to see that RSNDP is a special case of Bulk-SNDP with width at most $\max_i(r_i - 1)$. A disadvantage of Bulk-SNDP is that scenarios have to be explicitly listed, while the other models discussed specify failure scenarios implicitly. However, when connectivity requirements are small/constant, one can reduce to Bulk-SNDP by explicitly listing the failure sets.

---

[1] In the literature the term EC-SNDP and VC-SNDP are used to distinguish edge and vertex connectivity requirements. We use SNDP in place of EC-SNDP.

[2] If all edges are safe ($E = \mathcal{S}$), then $(p, 0)$-flex-connectivity is equivalent to p-edge-connectivity. Similarly, if all edges are unsafe ($E = \mathcal{U}$), then $(1, q - 1)$-flex-connectivity is equivalent to q-edge-connectivity.

While SNDP admits a 2-approximation, the approximability of network design in the preceding models is not well-understood. The known results mostly focus on two special cases: (i) the single pair case where there is only one pair (s, t) with a connectivity requirement and (ii) the spanning or global connectivity case when all pairs of vertices have identical connectivity requirement. Even in the single pair case, there are results that show that problems in the non-uniform models are hard to approximate to poly-logarithmic or almost-polynomial factors when the connectivity requirement is not bounded [6, 5]. Further, natural LP relaxations in some cases can also be shown to have large integrality gaps [16]. Motivated by these negative results and practical considerations, we focus our attention on Flex-SNDP when the max connectivity requirement p, q are small, and similarly on Bulk-SNDP when the width is small. Other network design problems with similar hardness results have admitted approximation ratios that depend on the max connectivity requirement (for example, VC-SNDP [12, 18, 38] and (s,t) case of Bulk-Robust [6]).

1.1 Our contribution

We are mainly motivated by Flex-SNDP and insights for it via Bulk-SNDP. We make two broad contributions. Our first set of results is on special cases of Flex-SNDP for which we obtain constant factor approximations. Our second contribution is a poly-logarithmic approximation for Flex-SNDP, Bulk-SNDP, and RSNDP when the requirements are small.

We use the terminology (p, q)-Flex-ST to refer to the single-pair problem with requirement (p, q). We use the term (p, q)-FGC to refer to the spanning/global-connectivity problem where all pairs of vertices have the (p, q)-flex-connectivity requirement (the term FGC is to be consistent with previous usage [3, 8]).

(p, q)-FGC. Adjiashvili et al. [3] considered $(1, 1)$-FGC and obtained a constant factor approximation that was subsequently improved to 2 by Boyd et al. [8]. [8] obtained several results for (p, q)-FGC including a 4-approximation for $(p, 1)$-FGC, a $(q + 1)$-approximation for $(1, q)$-FGC, and a $O(q \log n)$-approximation for (p, q)-FGC. The first non-trivial case of small p, q for which we did not know a constant factor is $(2, 2)$-FGC. We prove several results that, as a corollary, yield constant factor approximation for small values of p, q.

▶ **Theorem 1.** *For any $q \geq 0$ there is a $(2q + 2)$-approximation for $(2, q)$-FGC. For any $p \geq 1$ there is a $(2p + 4)$-approximation for $(p, 2)$-FGC, and a $(4p + 4)$-approximation for $(p, 3)$-FGC. Moreover, for all* even *$p \geq 2$ there is an $(6p + 4)$-approximation for $(p, 4)$-FGC.*

▶ Remark 2. In independent work Bansal et al. [7] obtained an $O(1)$-approximation for $(p, 2)$-FGC for any $p \geq 1$ (6 when p is even and 20 when p is odd). More broadly, they obtain constant factor approximations for a special class of augmentation problems and demonstrate several interesting applications.

(p, q)-Flex-ST. Adjiashivili et al. [4] considered $(1, q)$-Flex-ST and $(p, 1)$-Flex-ST and obtained several results. They described a q-approximation for $(1, q)$-Flex-ST and a $(p + 1)$-approximation for $(p, 1)$-Flex-ST; when p is a fixed constant they obtain a 2-approximation. Also implicit in [6] is an $O(q(p + q) \log n)$-approximation algorithm for (p, q)-Flex-ST that runs in $n^{O(p+q)}$-time. No constant factor approximation was known when $p, q \geq 2$ with $(2, 2)$-Flex-ST being the first non-trivial case. We prove a constant factor approximation for this and several more general settings via the following theorem.

▶ **Theorem 3.** *For all p, q where $(p+q) > pq/2$, there is an $O((p+q)^{O(p)})$-approximation algorithm for (p, q)-Flex-ST that runs in $n^{O(p+q)}$ time. In particular, there is an $O(1)$ approximation for $(p, 2)$ and $(2, q)$-Flex-ST when p, q are fixed constants.*

Flex-SNDP, Bulk-SNDP, and RSNDP. We show that these problems admit poly-logarithmic approximation algorithms when the width/connectivity requirements are small.

▶ **Theorem 4.** *There is a randomized algorithm that yields an $O(k^4 \log^7 n)$-approximation for Bulk-SNDP on instances with width at most k, and runs in expected polynomial time.*

▶ **Corollary 5.** *There is a randomized algorithm that yields an $O(q(p+q)^3 \log^7 n)$- approximation for Flex-SNDP when $(p_i, q_i) \leq (p, q)$ for all pairs (s_i, t_i), and runs in expected $n^{O(q)}$-time.*

▶ **Corollary 6.** *There is a randomized algorithm that yields an $O(k^4 \log^7 n)$-approximation for RSNDP where k is the maximum connectivity requirement, and runs in expected polynomial time.*

As far as we are aware of, no previous approximation algorithms were known for SNDP versions of flexible graph connectivity (with both $p, q \geq 2$) or bulk-robustness.

1.2 Overview of techniques and related work

Network design has substantial literature. We describe closely related work and results to put ours in context.

SNDP and related connectivity problems. SNDP is a canonical problem in network design for connectivity that captures many problems. We refer the reader to some older surveys [28, 36] on approximation algorithms for connectivity problems, and several recent papers with exciting progress on TSP and weighted Tree and Cactus augmentation. Frank's books is an excellent source for polynomial-time solvable exact algorithms [23]. For SNDP, the augmentation approach was pioneered in [41], and was refined in [25]. These led to $2H_k$ approximation where k is the maximum connectivity requirement. Jain's iterated rounding approach [32] obtained a 2-approximation. The nice structural results that underpin the algorithms for SNDP have been extended to *element* connectivity introduced in [33]; consequently, Elem-SNDP also admits a 2-approximation [21]. VC-SNDP has posed non-trivial technical challenges; the problem is not constant factor approximable when the maximum connectivity requirement is large [12]. In a breakthrough result, [18], Chuzhoy and Khanna gave an $O(k^3 \log n)$ approximation via a reduction to element connectivity, where k is the maximum connectivity. Nutov [38] improves this to an $O(k^2)$ approximation for the single-source VC-SNDP case; however, there has been no further progress in obtaining an $f(k)$-approximation for the general VC-SNDP problem.

Flexible Graph Connectivity. Flexible graph connectivity has been a topic of recent interest, although the model was introduced earlier in the context of a single pair [1]. Adjiashvili, Hommelsheim and Mühlenthaler [3] introduced FGC (which is the same as $(1, 1)$-FGC) and pointed out that it generalizes the well-known MST and 2-ECSS problems. Several approximation algorithms for various special cases of FGC and Flex-ST were obtained by Adjiashvili et al.[3] and Boyd et al. [8], as described in Section 1.1.

Adjiashvili et al. [5] also showed hardness results in the single pair setting. They prove that $(1, k)$-Flex-ST in directed graphs is at least as hard as directed Steiner tree which implies poly-logarithmic factor inapproximability [31]. They prove that $(k, 1)$-Flex-ST in directed graphs is at least as hard to approximate as directed Steiner forest (which has almost polynomial factor hardness [20]). The hardness results are when k is part of the input and large, and show that approximability of network design in this model is substantially different from the edge-connectivity model.

Bulk-Robust Network Design. This model was initiated in [6]. They obtained an $O(\log n + \log m)$ approximation for the Bulk-Robust spanning tree problem (as a special case of the more general matroid basis problem). The authors show that the *directed* single pair problem (Bulk-Robust shortest path) is very hard to approximate. The hardness reduction motivated the definition of width. The authors obtain an $O(k^2 \log n)$-approximation for Bulk-Robust shortest path via a nice reduction to the Set Cover problem and the use of the augmentation approach that we build upon here. For the special case of $k = 2$ the authors obtain an $O(1)$-approximation. Adjiashvili [2] showed that if the graph is planar then one can obtain an $O(k^2)$-approximation for both Bulk-Robust shortest path and spanning tree problems – he uses the augmentation approach from [6] and shows that the corresponding covering problem in each augmentation phase corresponds to a Set Cover problem that admits a constant factor approximation. As far as we are aware, there has not been any progress on the general setting beyond the spanning tree and shortest path cases.

Relative Network Design. This model was introduced in very recent work [19]. The authors obtain a 2-approximation for the spanning case via nice use of the iterated rounding technique even though the requirement function is not skew-supermodular. They also obtain a simple 2-approximation when the maximum requirement is 2. They obtain a $\frac{27}{4}$-approximation for the (s,t)-case when the maximum demand is 3.

Survivable Network Design for Group Connectivity. As we remarked, one part of this work builds on the recent framework of Chen et al. [17]. Their main motivation was to address the approximability of the survivable network design problem with group connectivity requirements. We refer the reader to [24, 30, 31, 14, 13, 17] and pointers to the extensive work on the approximability of these problems.

Our Techniques. As we remarked, the non-uniform models have been difficult to handle for existing algorithmic techniques. The structures that underpins the known algorithms for SNDP (primal-dual [41] and iterated rounding [32]) are skew-supermodularity of the requirement function and submodularity of the cut function in graphs. Since non-uniform models do not have such clean structural properties, these known techniques cannot be applied directly. Another technique for network design, based on several previous works, is augmentation. In the augmentation approach we start with an initial set of edges F_0 that partially satisfy the connectivity constraints. We then augment F_0 with a set F in the graph $G - F_0$; the augmentation is typically done to increase the connectivity by one unit for pairs that are not yet satisfied. We repeat this process in several stages until all connectivity requirements are met. The utility of the augmentation approach is that it allows one to reduce a higher-connectivity problem to a series of problems that solve a potentially simpler $\{0, 1\}$-connectivity problem. An important tool in this area is a 2-approximation for covering an *uncrossable* function (a formal definition is given in Section 3) [41].

In trying to use the augmentation approach for Flex-SNDP and its special cases, we see that the resulting functions are usually not uncrossable. To prove Theorems 1 and 3, we overcome this difficulty by decomposing the family of cuts to be covered in the augmentation problem into a sequence of cleverly chosen uncrossable subfamilies. Our structural results hold for certain range of values of p and q and hint at additional structure that may be available to exploit in future work. Boyd et al. also show a connection to *capacitated* network design (also implicitly in [5]) which has been studied in several works [25, 9, 10, 11]. This model generalizes standard edge connectivity by allowing each edge e to have an integer capacity $u_e \geq 1$. One can reduce capacitated network design to standard edge connectivity by replacing each edge e with u_e parallel edges, blowing up the approximation factor by $\max_e u_e$. Boyd et al. show that $(1, k)$-Flex-SNDP and $(k, 1)$-Flex-SNDP can be reduced to Cap-SNDP with maximum capacity k. While this reduction does not extend when $p, q \geq 2$, it provides a useful starting point that we exploit for Theorem 3.

We use a completely different algorithmic approach to prove Theorem 4. We rely on a recent novel framework of Chen, Laekhanukit, Liao, and Zhang [17] to tackle survivable network design in group connectivity setting. They used the seminal work of Räcke [39] on probabilistic approximation of capacitated graphs via trees, and the group Steiner tree rounding techniques of Garg, Konjevod and Ravi [24], and subsequent developments [27]. We adapt their ideas to handle the augmentation problem for Flex-SNDP and Bulk-SNDP. We refer the reader to Section 4 since the framework is technical.

Organization. Section 2 sets up the relevant background on the LP relaxations for Flex-SNDP and Bulk-SNDP. Section 3 outlines the proofs of Theorems 1 and 3. Section 4 outlines the proofs of Theorems 4 and the resulting corollaries 5 and 6. This paper combines and extends results from two preliminary versions; [16] for the first set of results discussed in Section 3, and [15] for the second set of results discussed in Section 4. A full version of this paper will be made publicly available in the near future.

2 Preliminaries

Throughout the paper we will assume that we are given an undirected graph $G = (V, E)$ along with a cost function $c : E \to \mathbb{R}_{\geq 0}$. When we say that H is a subgraph of $G = (V, E)$ we implicitly assume that H is an edge-induced subgraph, i.e. $H = (V, F)$ for some $F \subseteq E$. For any subset of edges $F \subseteq E$ and any set $S \subseteq V$ we use the notation $\delta_F(S)$ to denote the set of edges in F that have exactly one endpoint in S. We may drop F if it is clear from the context. For all discussion of flex-connectivity, we will let $\mathcal{S}$ denote the set of safe edges and $\mathcal{U}$ denote the set of unsafe edges.

Flex-SNDP LP Relaxation. We describe an LP relaxation for (p, q)-Flex-SNDP problem. Recall that we are given a set of h terminal pairs $(s_i, t_i) \subseteq V \times V$ and the goal is to choose a min-cost subset of the edges F such that in the subgraph $H = (V, F)$, s_i and t_i are (p, q)-flex-connected for any $i \in [h]$. Let $\mathcal{C} = \{S \subset V \mid \exists i \in [h] \text{ s.t. } |S \cap \{s_i, t_i\}| = 1\}$ be the set of all vertex sets that separate some terminal pair. For a set of edges F to be feasible for the given (p, q)-Flex-SNDP instance, we require that for all $S \in \mathcal{C}$, $|\delta_F(S) \setminus B| \geq p$ for any $B \subseteq \mathcal{U}$ with $|B| \leq q$. We can write cut covering constraints expressing this condition, but these constraints are not adequate by themselves. To improve this LP, we consider the connection to capacitated network design: we give each safe edge a capacity of $p + q$, each unsafe edge a capacity of p, and require $p(p + q)$ connectivity for the terminal pairs; it is not difficult to verify that this is a valid constraint. These two sets of constraints yield the following LP relaxation with variables $x_e \in [0, 1]$, $e \in E$.

$$\begin{aligned}
\min \sum_{e \in E} c(e) x_e & & \\
\text{subject to} \sum_{e \in \delta(S) - B} x_e \geq p & & S \in \mathcal{C}, B \subseteq \mathcal{U}, |B| \leq q \\
(p+q) \sum_{e \in \delta(S) \cap \mathcal{S}} x_e + p \sum_{e \in \delta(S) \cap \mathcal{U}} x_e \geq p(p+q) & & S \in \mathcal{C} \\
x_e \in [0,1] & & e \in E
\end{aligned}$$

The following lemma borrows ideas from [8, 10].

▶ **Lemma 7.** *The Flex-SNDP LP relaxation can be solved in $n^{O(q)}$ time. For (p,q)-FGC, it can be solved in polynomial time.*

Proof. We show a polynomial time separation oracle for the given LP. Suppose we are given some vector $x \in [0,1]^{|E|}$. We first check if the capacitated min-cut constraints are satisfied. This can be done in polynomial time by giving every safe edge a weight of $p+q$ and every unsafe edge a weight of p, and checking that the min-cut value is at least $p(p+q)$. If it is not, we can find the minimum cut and output the corresponding violated constraint. Suppose all capacitated constraints are satisfied. Then, for each $B \subseteq \mathcal{U}, |B| \leq q$ we remove B and check that for each $s,t \in T$, the s-t min-cut value in the graph $G - B$ with edge-capacities given by x is at least p. Since there are at most $n^{O(q)}$ such possible sets B, we get our desired separation oracle.

In the FGC case, if there is a remaining unsatisfied constraint, then there must must be some $S \subset V$ and some $B \subseteq \mathcal{U}$, $|B| \leq q$, such that $\sum_{e \in \delta(S)-B} x_e < p$. In particular, $\sum_{e \in \mathcal{S} \cap \delta(S) - B} x_e < p$ and $\sum_{e \in \mathcal{U} \cap \delta(S) - B} x_e < p$. We claim that the total weight (according to weights $(p+q)$ for safe edges and p for unsafe edges) going across $\delta(S)$ is at most $2p(p+q)$: at most $(p+q)p$ from $\mathcal{S} \cap (\delta(S) - B)$, at most p^2 from $\mathcal{U} \cap (\delta(S) - B)$, and at most pq from B. Recall that the min-cut of the graph according the weights has already been verified to be at least $p(p+q)$. Hence, any violated cut from the first set of constraints corresponds to 2-approximate min-cut. It is known via Karger's theorem that there are at most $O(n^4)$ 2-approximate min-cuts in a graph, and moreover they can also be enumerated in polynomial time [34, 35]. We can enumerate all 2-approximate min-cuts and check each of them to see if they are violated. To verify whether a candidate cut S is violated we consider the unsafe edges in $\delta(S) \cap \mathcal{U}$ and sort them in decreasing order of x_e value. Let B' be a prefix of this sorted order of size $\min\{q, |\delta(S) \cap \mathcal{U}|\}$. It is easy to see that that $\delta(S)$ is violated iff it is violated when $B = B'$. Thus, we can verify all candidate cuts efficiently. ◀

Bulk-SNDP LP Relaxation. We can similarly define an LP relaxation for the Bulk-SNDP problem. As above, we have a variable $x_e \in [0,1]$ for each edge $e \in E$.

$$\begin{aligned}
\min \sum_{e \in E} c(e) x_e & & \\
\text{subject to} \sum_{e \in \delta(S) \setminus F_j} x_e \geq 1 & & \forall (F_j, \mathcal{K}_j) \in \Omega, S \text{ separates a terminal pair in } \mathcal{K}_j \\
x_e \in [0,1] & &
\end{aligned}$$

Note that the LP has a separation oracle that runs in time polynomial in n and m: for each scenario $(F_j, \mathcal{K}_j)$, we can remove F_j from the graph and check that the minimum (u,v)-cut is at least 1 for each $(u,v) \in \mathcal{K}_j$.

Augmentation. The results of this paper rely on the augmentation framework. We first discuss Flex-SNDP. Suppose $G = (V, E), \{s_i, t_i\}_{i\in[h]}$ is an instance of (p, q)-Flex-SNDP. We observe that $(p, 0)$-Flex-SNDP instance can be solved via 2-approximation to EC-SNDP. Hence, we are interested in $q \geq 1$. Let F_1 be a feasible solution for the $(p, q-1)$-Flex-SNDP instance. This implies that for any cut S that separates a terminal pair we have $|\delta_{F_1\cap\mathcal{S}}(S)| \geq p$ or $|\delta_{F_1}(S)| \geq p + q - 1$. We would like to augment F_1 to obtain a feasible solution to satisfy the (p, q) requirement. Define a function $f : 2^{|V|} \to \{0, 1\}$ where $f(S) = 1$ iff (i) S separates a terminal pair and (ii) $|\delta_{F_1\cap\mathcal{S}}(S)| < p$ *and* $|\delta_{F_1}(S)| = p + q - 1$. We call S a *violated* cut with respect to F_1. Since F_1 satisfies $(p, q - 1)$ requirement, if $|\delta_{F_1\cap\mathcal{S}}| < p$ it must be the case that $|\delta_{F_1}(S)| \geq p + q - 1$. The following lemma is simple.

▶ **Lemma 8.** *Suppose $F_2 \subseteq E \setminus F_1$ is a feasible cover for f, that is, $\delta_{F_2}(S) \geq f(S)$ for all S. Then $F_1 \cup F_2$ is a feasible solution to (p, q)-Flex-SNDP.*

The augmentation problem is then to find a min-cost subset of edges to cover f in $G - F_1$. The key observation is that the augmentation problem does not distinguish between safe and unsafe edges and hence we can rely on traditional connectivity augmentation ideas. Note that if we instead tried to augment from $(p - 1, q)$ to (p, q)-flex-connectivity, we would still need to distinguish between safe and unsafe edges. The following lemma shows that the LP relaxation for the original instance provides a valid cut-covering relaxation for the augmentation problem.

▶ **Lemma 9.** *Let $x \in [0, 1]^{|E|}$ be a feasible LP solution for a given instance of (p, q)-Flex-Steiner. Let F_1 be a feasible solution that satisfies $(p, q - 1)$ requirements for the terminal. Then, for any violated cut $S \subseteq V$ in (V, F_1), we have $\sum_{e\in\delta(S)\setminus F_1} x_e \geq 1$.*

Adjiashvili et al [6] also define a corresponding augmentation problem for the bulk robust network design model as follows: given an instance to Bulk-SNDP, let $\Omega_\ell = \bigcup_{j\in[m]}\{(F, \mathcal{K}_j) : |F| \leq \ell \text{ and } F \subseteq F_j\}$. Let $H_\ell \subseteq E$ be a subset of edges that satisfy the constraints defined by the scenarios in Ω_ℓ. Then, a solution to the augmentation problem from $\ell - 1$ to ℓ is a set of edges H' such that $H_{\ell-1} \cup H'$ satisfies the constraints defined by scenarios in Ω_ℓ. It is not difficult to verify that any solution to the original instance Ω is also a solution to any of the augmentation problems, and any solution satisfying all scenarios in Ω_k also satisfies all scenarios in Ω, where k is the width of the Bulk-SNDP instance.

3 Uncrossability-Based Approximation Algorithms

In this section, we prove Theorem 1 and Theorem 3. Recall that we are given a graph $G = (V, E)$ with cost function on the edges $c : E \to \mathbb{R}_{\geq 0}$ and a partition of the edge set E into safe edges $\mathcal{S}$ and unsafe edges $\mathcal{U}$. In the (p, q)-FGC problem, our goal is to find the cheapest set of edges such that every cut has either p safe edges or $p + q$ total edges. The (p, q)-Flex-ST problem is similar, except that we are also given $s, t \in V$ as part of the input and we focus only on cuts separating s from t.

We start by providing some necessary background on uncrossable/ring families and submodularity of the cut function. We then prove a simple $O(q)$-approximation for $(2, q)$-FGC by directly applying existing algorithms for covering uncrossable functions. Next, we devise a framework for augmentation when the requirement function is not uncrossable. Finally, we prove our results for special cases of (p, q)-FGC and (p, q)-Flex-ST using this framework.

Uncrossable functions and families. Uncrossable functions are a general class of requirement functions that are an important ingredient in network design [41, 26, 28, 36].

▶ **Definition 10.** *A function $f : 2^V \to \{0,1\}$ is* uncrossable *if for every $A, B \subseteq V$ such that $f(A) = f(B) = 1$, one of the following is true: (i) $f(A \cup B) = f(A \cap B) = 1$, (ii) $f(A - B) = f(B - A) = 1$. A family of cuts $\mathcal{C} \subset 2^V$ is an uncrossable* family *if the indicator function $f_{\mathcal{C}} : 2^V \to \{0,1\}$ with $f(S) = 1$ iff $S \in \mathcal{C}$, is uncrossable.*

For a graph $G = (V, E)$, a requirement function $f : 2^V \to \{0,1\}$, and a subset of edges $A \subseteq E$, we say a set $S \subseteq V$ is violated with respect to A, f if $f(S) = 1$ and $\delta_A(S) = \emptyset$. The following important result gives a 2-approximation algorithm for the problem of covering an uncrossable requirement function.

▶ **Theorem 11** ([41])**.** *Let $G = (V, E)$ be an edge-weighted graph and let $f : 2^V \to \{0,1\}$ be an uncrossable function. Suppose there is an efficient oracle that for any $A \subseteq E$ outputs all the minimal violated sets of f with respect to A. Then there is an efficient 2-approximation for the problem of finding a minimum cost subset of edges that covers f.*

A special case of uncrossable family of sets is a *ring family*. We say that an uncrossable family $\mathcal{C} \subseteq 2^V$ is a ring family if the following conditions hold: (i) if $A, B \in \mathcal{C}$ and A, B properly intersect[3] then $A \cap B$ and $A \cup B$ are in $\mathcal{C}$ and (ii) there is a unique minimal set in $\mathcal{C}$. We observe that if $\mathcal{C}$ is an uncrossable family such that there is a vertex s contained in every $A \in \mathcal{C}$ then $\mathcal{C}$ is automatically a ring family. Theorem 11 can be strengthened for this case. There is an optimum algorithm to find a min-cost cover of a ring family – see [37, 38, 22].

In order to use Theorem 11 in the augmentation framework, we need to be able efficiently find all the minimal violated sets of the family. As above, we let F_1 denote a feasible solution for the $(p, q-1)$-Flex-SNDP instance. For any fixed p, q, we can enumerate all minimal violated sets in $n^{O(p+q)}$ time by trying all possible subsets of $p + q - 1$ edges in F_1. In the context of (p, q)-FGC, the total number of violated cuts in the augmentation problem is bounded by $O(n^4)$. See [8] and the proof of Lemma 7 for details.

For the following sections on (p, q)-FGC, we let $\mathcal{C}$ denote the family of violated cuts. Note that such families are symmetric, since $\delta(S) = \delta(V - S)$. For any two sets $A, B \in \mathcal{C}$, if $A \cup B = V$ then by symmetry, $V - A, V - B \in \mathcal{C}$. In this case, $V - A = B - A$ and $V - B = A - B$, so A and B uncross. Therefore, when proving uncrossability of A and B, we assume without loss of generality that $(A \cup B) \neq V$.

Submodularity and posimodularity of the cut function. It is well-known that the cut function of an undirected graph is symmetric and submodular. Submodularity implies that for all $A, B \subseteq V$, $|\delta(A)| + |\delta(B)| \geq |\delta(A \cap B)| + |\delta(A \cup B)|$. Symmetry and submodularity also implies posimodularity: for all $A, B \subseteq V$, $|\delta(A)| + |\delta(B)| \geq |\delta(A - B)| + |\delta(B - A)|$.

3.1 An $O(q)$-approximation for $(2, q)$-FGC

The following lemma shows that the augmentation problem for increasing flex-connectivity from $(2, q-1)$ to $(2, q)$, for any $q \geq 1$ corresponds to covering an uncrossable function.

▶ **Lemma 12.** *The set of all violated cuts when augmenting from $(2, q-1)$-FGC to $(2, q)$-FGC is uncrossable.*

---

[3] A, B properly intersect if $A \cap B \neq \emptyset$ and $A - B, B - A \neq \emptyset$.

The preceding lemma yields a $2(q+1)$-approximation for $(2,q)$-FGC as follows. We start with a 2-approximation for $(2,0)$-FGC that can be obtained by using an algorithm for 2-ECSS. Then for we augment in q-stages to go from a feasible solution to $(2,0)$-FGC to $(2,q)$-FGC. The cost of augmentation in each stage is at most OPT where OPT is the cost of an optimum solution to $(2,q)$-FGC. We can use the known 2-approximation algorithm in each augmentation stage since the family is uncrossable. Recall from Section 2 that the violated cuts can be enumerated in polynomial time, and hence the primal-dual 2-approximation for covering an uncrossable function can be implemented in polynomial-time. This leads to the claimed approximation and running time.

3.2 Identifying Uncrossable Subfamilies

We have seen that the augmentation problem from $(2,q-1)$-FGC to $(2,q)$-FGC leads to covering an uncrossable function. Boyd et al. [8] showed that augmenting from $(p,0)$-FGC to $(p,1)$-FGC also leads to an uncrossable function for any $p \geq 1$. However this approach fails for most cases of augmenting from $(p,q-1)$ to (p,q) (see [16] for examples). However, in certain cases, we can take a more sophisticated approach where we consider the violated cuts in a small number of stages. In each stage, we choose a subfamily of the violated cuts that is uncrossable. In such cases, we can obtain a $2k$-approximation for the augmentation problem, where k is the upper bound on the number of stages.

Suppose we want to augment from $(p,q-1)$ to (p,q) (for either FGC or the Flex-ST setting). Let $G=(V,E)$ be the original input graph, and let F be the set of edges we have already included. Recall that a cut $\emptyset \neq A \subsetneq V$ is violated iff $|\delta_F(A)| = p+q-1$, $|\delta_{F\cap\mathcal{S}}(A)| < p$, and in the Flex-ST case, A separates s from t. Instead of attempting to cover all violated sets at once, we do so in stages. In each stage we consider the violated cuts based on the number of safe edges. We begin by covering all violated sets with no safe edges, then with one safe edge, and iterate until all violated sets are covered. This is explained in Algorithm 1 below.

Algorithm 1 Augmenting from $(p,q-1)$ to (p,q) in stages.

$F' \leftarrow F$
for $i = 0,\dots,p-1$ **do**
  $\mathcal{C}_i \leftarrow \{S : S \text{ is violated and } |\delta_{F'\cap\mathcal{S}}(S)| = i\}$
  $F'_i \leftarrow$ approximation algorithm to cover cuts in $\mathcal{C}_i$
  $F' \leftarrow F' \cup F'_i$
end for
return F'

3.3 Approximating (p,q)-FGC for $q \leq 4$

In this section, we show that the above approach works to augment from $(p,q-1)$-FGC to (p,q)-FGC whenever $q \leq 3$ and also for $q=4$ when p is even. The only unspecified part Algorithm 1 is to cover cuts in $\mathcal{C}_i$ in the i'th stage. If we can prove that $\mathcal{C}_i$ forms an uncrossable family then we can obtain a 2-approximation in each stage. First, we prove a generic and useful lemma regarding cuts in $\mathcal{C}_i$.

For the remaining lemmas, we let $F_i \subseteq E$ denote the set of edges F' at the start of iteration i. In other words, F_i is a set of edges such that for all $\emptyset \neq A \subsetneq V$, if $|\delta_{F_i}(A)| = p+q-1$, then $|\delta_{F_i\cap\mathcal{S}}(A)| \geq i$.

▶ **Lemma 13.** *Fix an iteration $i \in \{0, \dots, p-1\}$. Let $\mathcal{C}_i$ be as defined in Algorithm 1. Then, if $A, B \in \mathcal{C}_i$ and*
1. $|\delta_{F_i}(A \cap B)| = |\delta_{F_i}(A \cup B)| = p + q - 1$*, or*
2. $|\delta_{F_i}(A - B)| = |\delta_{F_i}(B - A)| = p + q - 1$

then A and B uncross, i.e. $A \cap B, A \cup B \in \mathcal{C}_i$ or $A - B, B - A \in \mathcal{C}_i$.

Note that the preceding lemma holds for the high-level approach. Now we focus on cases where we can prove that $\mathcal{C}_i$ is uncrossable.

▶ **Lemma 14.** *Fix an iteration $i \in \{0, \dots, p-1\}$. Let $\mathcal{C}_i$ be as defined in Algorithm 1. Then, for $q \leq 3$, $\mathcal{C}_i$ is uncrossable.*

Proof. Suppose $A, B \subseteq V$ such that $\delta_{F_i}(A)$ and $\delta_{F_i}(B)$ both have exactly i safe and $p + q - 1 - i$ unsafe edges. Suppose for the sake of contradiction that they do not uncross. By Lemma 13, one of $\delta_{F_i}(A \cap B)$ and $\delta_{F_i}(A \cup B)$ must have at most $p + q - 2$ edges, and the same holds for $\delta_{F_i}(A - B)$ and $\delta_{F_i}(B - A)$. Without loss of generality, suppose $\delta_{F_i}(A \cap B)$ and $\delta_{F_i}(A - B)$ each have at most $p + q - 2$ edges. By the assumptions on F_i, they must both have at least p safe edges, hence they each have at most $q - 2$ unsafe edges. Note that $\delta_{F_i}(A) \subseteq \delta_{F_i}(A - B) \cup \delta_{F_i}(A \cap B)$, hence $\delta_{F_i}(A)$ can have at most $2(q - 2)$ unsafe edges. When $q \leq 3$, $2(q-2) < q$, which implies that $\delta_{F_i}(A)$ has strictly more than $p-1$ safe edges, a contradiction. Notice that $\delta_{F_i}(A) \subseteq \delta_{F_i}(B - A) \cup \delta_{F_i}(A \cup B)$, $\delta_{F_i}(B) \subseteq \delta_{F_i}(A - B) \cup \delta_{F_i}(A \cup B)$, and $\delta_{F_i}(B) \subseteq \delta_{F_i}(B - A) \cup \delta_{F_i}(A \cap B)$; therefore the same argument follows regardless of which pair of sets each have strictly less than $p + q - 2$ edges. ◀

▶ **Corollary 15.** *For any $p \geq 2$ there is a $(2p + 4)$-approximation for $(p, 2)$-FGC and a $(4p + 4)$-approximation for $(p, 3)$-FGC.*

Can we extend the preceding lemma for $q = 4$? It turns out that it does work when p is even but fails for odd $p \geq 3$.

▶ **Lemma 16.** *Fix an iteration $i \in \{0, \dots, p-2\}$. Let $\mathcal{C}_i$ be as defined in Algorithm 1. Then, for $q = 4$, $\mathcal{C}_i$ is uncrossable. Furthermore, if p is an* even *integer, $\mathcal{C}_{p-1}$ is uncrossable.*

The preceding lemma leads to a $(6p + 4)$-approximation for $(p, 4)$-FGC when p is even by augmenting from a feasible solution to $(p, 3)$, since we pay an additional cost of $2p \cdot OPT$. The preceding lemma also shows that the bottleneck for odd p is in covering $\mathcal{C}_{p-1}$. It may be possible to show that $\mathcal{C}_{p-1}$ separates into a constant number of uncrossable families leading to an $O(p)$-approximation for $(p, 4)$-FGC for all p. The first non-trivial case is when $p = 3$.

3.4 An $O(1)$-Approximation for Flex-ST

In this section, we provide a constant factor approximation for (p, q)-Flex-ST for all fixed p, q that satisfy $2(p + q) > pq$. We follow the general approach outlined in 3.2 with some modifications. In particular, we start with a stronger set of edges F than in the FGC case above. Let $E' \subseteq E$ denote a feasible solution to $(p, q - 1)$-Flex-ST.

Recall from Section 1 the capacitated network design problem, in which each edge has an integer capacity $u_e \geq 1$. Consider an instance of the $(p(p + q))$-Cap-ST problem on G where every safe edge is given a capacity of $p + q$ and every unsafe edge is given a capacity of p. Our goal is to find the cheapest set of edges that support a flow of $(p(p + q))$ from s to t. It is easy to see that any solution to (p, q)-Flex-ST is also a feasible solution for this capacitated problem: every s-t cut either has at least p safe edges or at least $p + q$ total edges, and either case gives a capacity of at least $p(p + q)$. As mentioned in Section 1, there exists a

$2\max_e(u_e) = 2(p+q)$ approximation for this problem. Let $E'' \subseteq E$ be such a solution, and note that $\text{cost}(E'') \leq 2(p+q) \cdot OPT$, where OPT denotes the cost of an optimal solution to (p,q)-Flex-ST.

Let $F = E' \cup E''$. We redefine $\mathcal{C}$ to limit ourselves to the set of violated cuts containing s, i.e. $\mathcal{C} = \{A \subset V : s \in A, t \notin A, |\delta_{F\cap\mathcal{S}}(A)| < p, |\delta_{F\cap\mathcal{U}}(A)| = p+q-1\}$. By symmetry, it suffices to only consider cuts containing s, since covering a set also covers its complement. Following the discussion in Section 3.2, we use Algorithm 1 to cover violated cuts in stages based on the number of safe edges. However, unlike the spanning case, the sets $\mathcal{C}_i$ are not uncrossable in the single pair setting, even for $(2,2)$-Flex-ST. In this case, we aim to further partition $\mathcal{C}_i$ into subfamilies that we can cover efficiently.

For the remaining lemmas, we let $F_i \subseteq E$ denote the set of edges F' at the start of iteration i. In other words, F_i is a set of edges such that for all cuts A separating s from t, if $|\delta_{F_i}(A)| = p+q-1$, then $|\delta_{F_i\cap\mathcal{S}}(A)| \geq i$. We begin with a structural lemma.

▶ **Lemma 17.** *Fix an iteration $i \in \{0,\ldots,p-1\}$. Let $\mathcal{C}_i$ be as defined in Algorithm 1. Let $A, B \in \mathcal{C}_i$. Then, either*
1. *$A \cup B, A \cap B \in \mathcal{C}_i$, i.e. A and B uncross, or*
2. *$\max(\delta_{F_i\cap\mathcal{S}}(A\cap B), \delta_{F_i\cap\mathcal{S}}(A\cup B)) \geq p$.*

Consider a flow network on the graph (V, F_i) with safe edges given a capacity of $(p+q)$ and unsafe edges given a capacity of p. Since $E'' \subseteq F_i$, F_i satisfies the $p(p+q)$-Cap-ST requirement. Therefore, the minimum capacity s-t cut and thus the maximum s-t flow value is at least $p(p+q)$. Since capacities are integral, there is some integral max flow f. By flow decomposition, we can decompose f into a set $\mathcal{P}$ of $|f|$ paths, each carrying a flow of 1, and we can find $\mathcal{P}$ in polynomial time.

For each $\mathcal{Q} \subseteq \mathcal{P}$ where $|\mathcal{Q}| = i$, we define a subfamily of violated cuts $\mathcal{C}_i^{\mathcal{Q}}$ as follows. Let $\mathcal{Q} = P_1, \ldots, P_i$. Then, $A \in \mathcal{C}_i$ is in $\mathcal{C}_i^{\mathcal{Q}}$ iff there exist distinct edges $e_1, \ldots, e_i \in \mathcal{S}$ satisfying:
1. $\forall j \in [i]$, $\delta_{F_i}(A) \cap P_j = \{e_j\}$,
2. $\delta_{F_i\cap\mathcal{S}}(A) = \{e_1, \ldots, e_i\}$.

Informally, $\mathcal{C}_i^{\mathcal{Q}}$ is the set of all violated cuts that intersect the paths of $\mathcal{Q}$ exactly once and on a distinct safe edge each.

▶ **Lemma 18.** *Suppose $pq < 2p + 2q$. If $A \in \mathcal{C}_i$, then there exists some $\mathcal{Q} \subseteq \mathcal{P}$, $|\mathcal{Q}| = i$, such that $A \in \mathcal{C}_i^{\mathcal{Q}}$.*

The above lemma shows that $\cup_{\mathcal{Q}\subseteq\mathcal{P},|\mathcal{Q}|=i}\mathcal{C}_i^{\mathcal{Q}} = \mathcal{C}_i$. Therefore, it suffices to cover each $\mathcal{C}_i^{\mathcal{Q}}$.

▶ **Lemma 19.** *For any $\mathcal{Q} \subseteq \mathcal{P}$, $|\mathcal{Q}| = i$, $\mathcal{C}_i^{\mathcal{Q}}$ is a ring family.*

We combine the above lemmas to obtain a constant factor approximation for covering $\mathcal{C}_i$.

▶ **Lemma 20.** *Suppose $2(p+q) > pq$ and p, q are fixed. Then, there exists an algorithm that runs in $n^{O(p+q)}$ time to cover all cuts in $\mathcal{C}_i$ with cost at most $\binom{p^2+2pq}{i} \cdot OPT$.*

The above lemma gives us Theorem 3 as a corollary. At the beginning of each augmentation step, before running Algorithm 1, we compute a solution to the $(p(p+q))$-Cap-ST problem, which we can do with cost at most $(p+q) \cdot OPT$. Summing over q augmentation iterations gives us the desired $(p+q)^{O(p)}$ approximation ratio.

▶ Remark 21. The approximation factor in Theorem 3 can be optimized slightly. For example, the algorithm we describe gives a 5-approximation for $(2,2)$-Flex-ST. We omit the details of this optimization in this paper and instead focus on showing constant factor for fixed p, q.

4 Approximating the Augmentation Problem for (p, q)-Flex-SNDP

In this section, we prove Theorem 4 and the resulting Corollaries 5 and 6. We begin with some background on Räcke's capacity-based probabilistic tree embeddings and Tree Rounding algorithms for Group Steiner tree. We then present the algorithm and analysis for Bulk-SNDP.

4.1 Räcke Tree Embeddings

The results in this section use Räcke's capacity-based probabilistic tree embeddings. We borrow the notation from [17]. Given $G = (V, E)$ with capacity $x : E \to \mathbb{R}^+$ on the edges, a capacitated tree embedding of G is a tree $\mathcal{T}$, along with two mapping functions $\mathcal{M}_1 : V(\mathcal{T}) \to V(G)$ and $\mathcal{M}_2 : E(\mathcal{T}) \to 2^{E(G)}$ that satisfy some conditions. $\mathcal{M}_1$ maps each vertex in $\mathcal{T}$ to a vertex in G, and has the additional property that it gives a one-to-one mapping between the leaves of $\mathcal{T}$ and the vertices of G. $\mathcal{M}_2$ maps each edge $(a, b) \in E(\mathcal{T})$ to a path in G between $\mathcal{M}_1(a)$ and $\mathcal{M}_1(b)$. For notational convenience we view the two mappings as a combined mapping $\mathcal{M}$. For a vertex $u \in V(G)$ we use $\mathcal{M}^{-1}(u)$ to denote the leaf in $\mathcal{T}$ that is mapped to u by $\mathcal{M}_1$. For an edge $e \in E(G)$ we use $\mathcal{M}^{-1}(e) = \{f \in E(\mathcal{T}) \mid e \in \mathcal{M}_2(f)\}$. It is sometimes convenient to view a subset $S \subseteq V(G)$ both as vertices in G and also corresponding leaves of $\mathcal{T}$.

The mapping $\mathcal{M}$ induces a capacity function $y : E(\mathcal{T}) \to \mathbb{R}_+$ as follows. Consider $f = (a, b) \in E(\mathcal{T})$. $\mathcal{T} - f$ induces a partition (A, B) of $V(T)$ which in turn induces a partition/cut (A', B') of $V(G)$ via the mapping $\mathcal{M}$: A' is the set of vertices in G that correspond to the leaves in A and similarly B'. We then set $y(f) = \sum_{e\in\delta(A')} x(e)$, in other words $y(f)$ is the capacity of cut (A', B') in G. The mapping also induces loads on the edges of G. For each edge $e \in G$, we let $\text{load}(e) = \sum_{f\in E(\mathcal{T}):e\in M(f)} y(f)$. The relative load or *congestion* of e is $\text{rload}(e) = \text{load}(e)/x(e)$. The congestion of G with respect to a tree embedding $(\mathcal{T}, \mathcal{M})$ is defined as $\max_{e\in E(G)} \text{rload}(e)$. Given a probabilistic distribution $\mathcal{D}$ on trees embeddings of (G, x) we let $\beta_{\mathcal{D}} = \max_{e\in E(G)} \mathbf{E}_{(\mathcal{T},\mathcal{M})\sim\mathcal{D}}\, \text{rload}(e)$ denote the maximum expected congestion. Räcke showed the following fundamental result on probabilistic embeddings of a capacitated graph into trees.

▶ **Theorem 22** ([39]). *Given a graph G and $x : E(G) \to \mathbb{R}^+$, there exists a probability distribution $\mathcal{D}$ on tree embeddings such that $\beta_{\mathcal{D}} = O(\log |V(G)|)$. All trees in the support of $\mathcal{D}$ have height at most $O(\log(nC))$, where C is the ratio of the largest to smallest capacity in x. Moreover, there is randomized polynomial-time algorithm that can sample a tree from the distribution $\mathcal{D}$.*

In the rest of the paper we use β to denote the guarantee provided by the preceding theorem where $\beta = O(\log n)$ for a graph on n nodes. In order to use these probabilistic embeddings to route flow, we need the following corollary, where we use $\text{maxflow}^z_H(A, B)$ to denote the maxflow between two disjoint vertex subsets A, B in a capacitated graph H with capacities given by $z : E(H) \to \mathbb{R}_+$.

▶ **Corollary 23.** *Let $\mathcal{D}$ be the distribution guaranteed in Theorem 22. Let $A, B \in V(G)$ be two disjoint sets. Then*

(i) *for any tree $(\mathcal{T}, \mathcal{M})$ in $\mathcal{D}$, $\text{maxflow}^x_G(A, B) \le \text{maxflow}^y_{\mathcal{T}}(\mathcal{M}^{-1}(A), \mathcal{M}^{-1}(B))$ and*

(ii) $\frac{1}{\beta}\, \mathbf{E}_{(\mathcal{T},\mathcal{M})\sim\mathcal{D}}[\text{maxflow}^y_{\mathcal{T}}(\mathcal{M}^{-1}(A), \mathcal{M}^{-}(B))] \le \text{maxflow}^x_G(A, B)$.

4.2 Group Steiner Tree, Set Connectivity and Tree Rounding

The group Steiner tree problem was introduced in [40] and studied in approximation by Garg, Konjevod and Ravi [24]. The input is an edge-weighted graph $G = (V, E)$, a root vertex $r \in V$, and k groups $S_1, S_2, \ldots, S_k$ where each $S_i \subseteq V$. The goal is to find a min-weight subgraph H of G such there is a path in H from r to each group S_i (that is, to some vertex in S_i). The approximability of this problem has attracted substantial attention. Garg et al. [24] described a randomized algorithm to round a fractional solution to a cut-based LP relaxation when G is a tree – it achieves a $O(\log n \log k)$-approximation.

Set Connectivity is a generalization of group Steiner tree problem. Here we are given pairs of sets $(S_1, T_1), (S_2, T_2), \ldots, (S_k, T_k)$ and the goal is to find a min-cost subgraph H such that there is an (S_i, T_i) path in H for each i. Chalermsook, Grandoni and Laekhanukit [13] studied Survivable Set Connectivity problem, motivated by earlier work in [29]. Here each pair (S_i, T_i) has a connectivity requirement r_i which implies that one seeks r_i edge-disjoint paths between S_i and T_i in the chosen subgraph H; [13] obtained a bicriteria-approximation via Räcke tree and group Steiner tree rounding. The recent work of Chen et al [17] uses related but more sophisticated ideas to obtain the first true approximation for this problem. They refer to the problem as Group Connectivity problem and obtain an $O(r^3 \log r \log^7 n)$-approximation where $r = \max_i r_i$ connectivity requirement (see [17] for more precise bounds).

Oblivious tree rounding. In [13] a randomized oblivious algorithm based on the group Steiner tree rounding from [24] is described. This is useful since the sets to be connected during the course of their algorithm are implicitly generated. We encapsulate their result in the following lemma. The tree rounding algorithm in [13, 17] is phrased slightly differently since they combine aspects of group Steiner rounding and the congestion mapping that comes from Räcke trees. We separate these two explicitly to make the idea more transparent. We refer to the algorithm from the lemma below as TreeRounding.

▶ **Lemma 24** ([13, 17]). *Consider an instance of Set Connectivity on an n-node tree $T = (V, E)$ with height h and let $x : E \to [0, 1]$. Suppose $A, B \subseteq V$ are disjoint sets and suppose $K \subseteq E$ such that x restricted to K supports a flow of $f \leq 1$ between A and B. There is a randomized algorithm that is oblivious to A, B, K (hence depends only on x and value f) that outputs a subset $E' \subseteq E$ such that (i) The probability that $E' \cap K$ connects A to B is at least a fixed constant ϕ and (ii) For any edge $e \in E$, the probability that $e \in E'$ is $\min\{1, O(\frac{1}{f} h \log^2 n) x(e)\}$.*

4.3 Rounding Algorithm for the Augmentation Problem

We adapt the algorithm and analysis in [17] to Bulk-SNDP. Let β be the expected congestion given by Theorem 22. Consider an instance of Bulk-SNDP specified by a graph $G : (V, E)$ with cost function $c : E \to \mathbb{R}_{\geq 0}$ and a set of scenarios $\Omega = \{(F_j, \mathcal{K}_j) : j \in [m]\}$. Assume we have a partial solution H satisfying all scenarios in $\Omega_{\ell-1}$. We augment H to satisfy scenarios in Ω_ℓ.

We start by obtaining a solution $\{x_e\}_{e \in E \setminus H}$ for the LP relaxation. Let $E' = E \setminus H$. We define LARGE $= \{e \in E' : x_e \geq \frac{1}{4\ell\beta}\}$, and SMALL $= \{e \in E' : x_e < \frac{1}{4\ell\beta}\}$. The LP has paid for each $e \in$ LARGE a cost of at least $c(e)/(4\ell\beta)$, hence adding all of them to H will cost $O(\ell\beta \cdot \mathrm{OPT_{LP}})$. If LARGE $\cup H$ is a feasible solution to the augmentation problem, then we are done since we obtain a solution of cost $O(\ell \log n \cdot \mathrm{OPT_{LP}})$. Thus, the interesting case is when LARGE $\cup H$ is *not* a feasible solution.

Following [17] we employ a Räcke tree based rounding. A crucial step is to set up a capacitated graph appropriately. We can assume, with a negligible increase in the fractional cost, that for each edge $e \in E'$, $x(e) = 0$ or $x(e) \geq \frac{1}{n^3}$; this can be ensured by rounding down to 0 the fractional value of any edge with very small value, and compensating for this loss by scaling up the fractional value of the other edges by a factor of $(1 + 1/n)$. It is easy to check that the new solution satisfies the cut covering constraints, and we have only increased the cost of the fractional solution by a $(1 + 1/n)$-factor. In the subsequent steps we can ignore edges with $x_e = 0$ and assume that there are no such edges.

Consider the original graph $G = (V, E)$ where we set a capacity for each $e \in E$ as follows. If $e \in \text{LARGE} \cup H$ we set $\tilde{x}_e = \frac{1}{4\ell\beta}$. Otherwise we set $\tilde{x}_e = x_e$. Since the ratio of the largest to smallest capacity is $O(n^3)$, the height of any Räcke tree for G with capacities $\tilde{x}$ is at most $O(\log n)$. Then, we repeatedly sample Räcke trees. For each tree, we sample edges by the rounding algorithm given by Chalermsook et al in [13] (see Section 4.2 for details). A formal description of the algorithm is provided below where t' and t are two parameters that control the number of trees sampled and the number of times we run the tree rounding algorithm in each sampled tree. We will analyze the algorithm by setting both t and t' to $\Theta(\ell \log n)$.

Algorithm 2 Approximating the Bulk-SNDP Augmentation Problem from $\ell - 1$ to ℓ.

$H \leftarrow$ partial solution satisfying scenarios in Ω_ℓ
$\{x\}_{e\in E} \leftarrow$ fractional solution to the LP
$\text{LARGE} \leftarrow \{e \in E' : x_e \geq \frac{1}{4\ell\beta}\}$
$\text{SMALL} \leftarrow \{e \in E' : x_e < \frac{1}{4\ell\beta}\}$
$H \leftarrow H \cup \text{LARGE}$
if H is a feasible solution satisfying scenarios in $\Omega_{\ell+1}$ **then return** H
else

$$\tilde{x}_e \leftarrow \begin{cases} \frac{1}{4\ell\beta} & e \in H \\ x_e & \text{otherwise} \end{cases}$$

end if
$\mathcal{D} \leftarrow$ Räcke tree distribution for $(G, \tilde{x})$
for $i = 1, \dots t'$ **do**
  Sample a tree $(\mathcal{T}, \mathcal{M}, y) \sim \mathcal{D}$
  for $j = 1, \dots, t$ **do**
    $H' \leftarrow$ output of oblivious TreeRounding algorithm on $(G, \mathcal{T})$
    $H \leftarrow H \cup \mathcal{M}(H')$
  end for
end for
return H

4.4 Analysis

For the remainder of of this analysis, we denote as H the partial solution after buying edges in LARGE. We will assume, following earlier discussion, that H does not satisfy all requirements specified by scenarios in Ω_ℓ. This implies that there must be some F such that $|F| \leq \ell$, $F \subseteq F_i$ for some $i \in [m]$, and $\exists (u, v) \in \mathcal{K}_i$ such that u and v are disconnected in $(V, H \setminus F)$. Since H satisfies all scenarios in $\Omega_{\ell-1}$, it must be the case that F has exactly ℓ edges. We call such an F a violating set. There are at most $\binom{|H|}{\ell}$ violating edge sets, and since $|H| \leq n^2$, this is upper bounded by $O(n^{2\ell})$. We say that a set of edges $H' \subseteq E \setminus H$ is a feasible augmentation for violating edge set F if $\forall i \in [m]$ such that $F \subseteq F_i$, $\forall (u, v) \in \mathcal{K}_i$, there is a path from u to v in $(H \cup H') \setminus F$. The following is a simple observation.

▷ Claim 25. $H' \subseteq E \setminus H$ is a feasible solution to the augmentation problem iff for each violating edge set F, H' is a feasible augmentation for F.

The preceding observation allows us to focus on a fixed violating edge set F, and ensure that the algorithm outputs a set H' that is a feasible augmentation for F with high probability. We observe that the algorithm is oblivious to F. Thus, if we obtain a high probability bound for a fixed F, since there are $O(n^{2\ell})$ violating edge sets, we can use the union bound to argue that H' is feasible solution for *all* violating edge sets. For the remainder of this section, until we do the final cost analysis, we work with a fixed violating edge set F. For ease of notation, we let $\mathcal{K}_F = \bigcup_{i\in[m]:F\subseteq F_i} \mathcal{K}_i$ be the set of terminal pairs that need to be connected in $(H \cup H') \setminus F$.

Consider a tree $(\mathcal{T}, \mathcal{M}, y)$ in the Räcke distribution for the graph G with capacities $\tilde{x}$. We let $\mathcal{M}^{-1}(F)$ denote the set of all tree edges corresponding to edges in F, i.e. $\mathcal{M}^{-1}(F) = \cup_{e\in F}\mathcal{M}^{-1}(e)$. We call $(\mathcal{T}, \mathcal{M}, y)$ *good* with respect to F if $y(\mathcal{M}^{-1}(F)) \leq \frac{1}{2}$; equivalently, F blocks a flow of at most $\frac{1}{2}$ in $\mathcal{T}$.

▶ **Lemma 26.** *For a violating edge set F, a randomly sampled Räcke tree $(\mathcal{T}, \mathcal{M}, y)$ is good with respect to F with probability at least $\frac{1}{2}$.*

Given the preceding lemma, a natural approach is to sample a good tree $\mathcal{T}$ and hope that $\mathcal{T} \setminus M^{-1}(F)$ still has good flow between each terminal pair. However, since we rounded down all edges in LARGE $\cup\, H$, it is possible that $\mathcal{M}^{-1}(F)$ contains an edge whose removal would disconnect a terminal pair in $\mathcal{T}$, even if $\mathcal{T}$ is good. See [17] for a more detailed discussion and example.

We note that our goal is to find a set of edges $H' \subseteq E$ such that each terminal pair in $\mathcal{K}_F$ has a path in $(H' \cup H) \setminus F$; these paths must exist in the original graph, even if they do not exist in the tree. Therefore, instead of looking directly at paths in $\mathcal{T}$, we focus on obtaining paths through components that are already connected in $(V(G), H \setminus F)$. The rest of the argument is to show that sufficiently many iterations of TreeRounding on any good tree $\mathcal{T}$ for F will yield a feasible set H' for F.

4.5 Shattered Components, Set Connectivity and Rounding

Let $\mathbb{Q}_F$ be the set of connected components in the subgraph induced by $H \setminus F$. We use vertex subsets to denote components. Let $\mathcal{T}$ be a good tree for F. We say that a connected component $Q \in \mathbb{Q}_F$ is *shattered* if it is disconnected in $\mathcal{T} \setminus \mathcal{M}^{-1}(F)$, else we call it *intact*. For each $(u, v) \in \mathcal{K}_F$, let $Q_u \in \mathbb{Q}_F$ be the component containing u, and $Q_v \in \mathbb{Q}_F$ be the component containing v. Note that Q_u may be the same as Q_v for some $(u, v) \in \mathcal{K}_F$, but if F is a violating edge set then there is at least one pair $(u, v) \in \mathcal{K}_F$ such that $Q_u \neq Q_v$. Now, we define a Set Connectivity instance that is induced by F and $\mathcal{T}$. Consider two disjoint vertex subsets $A, B \subset V$. We say that (A, B) partitions the set of shattered components if each shattered component Q is fully contained in A or fully contained in B. Formally let

$$Z_F = \{(A \cup Q_u, B \cup Q_v) : (A, B) \text{ partitions the shattered components}, (u, v) \in \mathcal{K}_F\}.$$

In other words, Z_F is set of all partitions of shattered components that separate some pair $(u, v) \in \mathcal{K}_F$. Since the leaves of $\mathcal{T}$ are in one to one correspondence with $V(G)$ we can view Z_F as inducing a Set Connectivity instance in $\mathcal{T}$; technically we need to consider the pairs $\{(\mathcal{M}^{-1}(A), \mathcal{M}^{-1}(B)) \mid (A, B) \in Z_F\}$; however, for simplicity we conflate the leaves of $\mathcal{T}$ with $V(G)$. We claim that it suffices to find a feasible solution that connects the pairs defined by Z_F in the tree $\mathcal{T}$.

▶ **Lemma 27.** *Let $E' \subseteq \mathcal{T} \setminus \mathcal{M}^{-1}(F)$. Suppose there exists a path in $E' \subseteq \mathcal{T} \setminus \mathcal{M}^{-1}(F)$ connecting A to B for all $(A, B) \in Z_F$. Then, there is an u-v path for each $(u, v) \in \mathcal{K}_F$ in $(\mathcal{M}(E') \cup H) \setminus F$.*

Routing flow. We now argue that $(\mathcal{T}, \mathcal{M}, y)$ routes sufficient flow for each pair in Z_F without using the edges in $\mathcal{M}^{-1}(F)$; in other words y is fractional solution (modulo a scaling factor) to the Set Connectivity instance Z_F in the graph/forest $\mathcal{T} \setminus \mathcal{M}^{-1}(F)$. We can then appeal to TreeRounding lemma to argue that it will connect the pairs in Z_F without using any edges in F.

▶ **Lemma 28.** *Let $(A, B) \in Z_F$. Let $S \subset V_\mathcal{T}$ such that $A \subseteq S$ and $B \subseteq V_\mathcal{T} \setminus S$. Then $y(\delta_{\mathcal{T} \setminus \mathcal{M}^{-1}(F)}(S)) \geq \frac{1}{4\ell\beta}$.*

Bounding Z_F. A second crucial property is a bound on $|Z_F|$, the number of pairs in the Set Connectivity instance induced by F and a good tree $\mathcal{T}$ for F.

▶ **Lemma 29.** *For a good tree $\mathcal{T}$, $|Z_F| \leq 2^{2\ell\beta}|\mathcal{K}_F|$.*

4.6 Correctness and Cost

The following two lemmas show that by taking a union bound over all violating edge sets F and applying the Tree Rounding lemma 24, one can show that the algorithm outputs a feasible augmentation solution with probability at least $\frac{1}{2}$.

▶ **Lemma 30.** *Suppose $\mathcal{T}$ is good for a violating edge set F. Then after $t = O(\ell log n)$ rounds of TreeRounding with flow parameter $\frac{1}{4\ell\beta}$, the probability that H' is* not *a feasible augmentation for F is at most $(1-\phi)^t |Z_F| \leq 1/4$.*

▶ **Lemma 31.** *The algorithm outputs a solution H' such that $H \cup H'$ is a feasible augmentation to the given instance with probability at least $\frac{1}{2}$.*

Now we analyze the expected cost of the edges output by the algorithm for augmentation with respect to OPT_{LP}, the cost of the fractional solution.

▶ **Lemma 32.** *The total expected cost of the algorithm is $O(\ell^3 \log^7 n) \cdot OPT_{LP}$.*

Combining the correctness and cost analysis we obtain the following.

▶ **Lemma 33.** *There is a randomized $O(\ell^3 \log^7 n)$-approximation algorithm for the Bulk-SNDP Augmentation problem from $\ell - 1$ to ℓ. The algorithm runs in time polynomial in n and α, where α is the amount of time it takes to solve the LP.*

To prove Theorem 4, we start with a solution from $\ell = 0$ and iteratively solve k augmentation problems. Since the LP for Bulk-SNDP can be solved in polynomial time (see Section 2), we obtain a polynomial time $O(k^4 \log^7 n)$-approximation algorithm.

For Flex-SNDP, recall that $(p, 0)$-Flex-SNDP is equivalent to EC-SNDP where every terminal pair has connectivity requirement $r_i = p$. Therefore, we can start with a 2-approximate solution to $(p, 0)$-Flex-SNDP and apply Lemma 33 q times. In this case, the maximum width is $p + q$, so we get an overall approximation ratio of $O(q(p + q)^3 \log^7 n)$. Recall from Section 2 that the LP can be solved in $n^{O(q)}$ time, giving us Corollary 5.

Finally, for Relative SNDP, there is an LP relaxation described in [19] that can be solved in polynomial time, even when k is not fixed. We can modify Algorithm 2 for RSNDP by solving this LP relaxation instead and following the same rounding algorithm. This, along with the reduction to Bulk-SNDP discussed in Section 1, completes the proof of Corollary 6.

References

1 David Adjiashvili. Fault-tolerant shortest paths – Beyond the uniform failure model, 2013. doi:10.48550/arXiv.1301.6299.

2 David Adjiashvili. Non-uniform robust network design in planar graphs. *arXiv preprint*, 2015. arXiv:1504.05009.

3 David Adjiashvili, Felix Hommelsheim, and Moritz Mühlenthaler. Flexible graph connectivity. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 13–26. Springer, 2020.

4 David Adjiashvili, Felix Hommelsheim, and Moritz Mühlenthaler. Flexible graph connectivity. *Mathematical Programming*, 192(1):409–441, 2022.

5 David Adjiashvili, Felix Hommelsheim, Moritz Mühlenthaler, and Oliver Schaudt. Fault-Tolerant Edge-Disjoint s-t Paths – Beyond Uniform Faults. In Artur Czumaj and Qin Xin, editors, *18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022)*, volume 227 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SWAT.2022.5.

6 David Adjiashvili, Sebastian Stiller, and Rico Zenklusen. Bulk-robust combinatorial optimization. *Mathematical Programming*, 149(1):361–390, 2015.

7 Ishan Bansal, Joseph Cheriyan, Logan Grout, and Sharat Ibrahimpur. Improved approximation algorithms by generalizing the primal-dual method beyond uncrossable functions, 2022. To appear in Proc. of ICALP 2023. doi:10.48550/arXiv.2209.11209.

8 Sylvia Boyd, Joseph Cheriyan, Arash Haddadan, and Sharat Ibrahimpur. Approximation algorithms for flexible graph connectivity. *Mathematical Programming*, pages 1–24, 2023. Preliminary version appeared in Proc. of FSTTCS 2021. doi:10.1007/s10107-023-01961-5.

9 R. D. Carr, L. K. Fleischer, V. J. Leung, and C. A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 106–115. Society for Industrial and Applied Mathematics, 2000.

10 Deeparnab Chakrabarty, Chandra Chekuri, Sanjeev Khanna, and Nitish Korula. Approximability of capacitated network design. *Algorithmica*, 72(2):493–514, 2015.

11 Deeparnab Chakrabarty, Ravishankar Krishnaswamy, Shi Li, and Srivatsan Narayanan. Capacitated network design on undirected graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 71–80. Springer, 2013.

12 Tanmoy Chakraborty, Julia Chuzhoy, and Sanjeev Khanna. Network design for vertex connectivity. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 167–176, 2008.

13 Parinya Chalermsook, Fabrizio Grandoni, and Bundit Laekhanukit. On survivable set connectivity. In *Proceedings of the 2015 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 25–36. Society for Industrial and Applied Mathematics, 2015. doi:10.1137/1.9781611973730.3.

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

15 Chandra Chekuri and Rhea Jain. Approximating flexible graph connectivity via räcke tree based rounding, 2022. doi:10.48550/arXiv.2211.08324.

16 Chandra Chekuri and Rhea Jain. Augmentation based approximation algorithms for flexible network design, 2022. doi:10.48550/arXiv.2209.12273.

17 Qingyun Chen, Bundit Laekhanukit, Chao Liao, and Yuhao Zhang. Survivable network design revisited: Group-connectivity, 2022. Full version of paper in Proceedings of IEEE FOCS 2022. doi:10.48550/arXiv.2204.13648.

18 J. Chuzhoy and S. Khanna. An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design. *Theory of Computing*, 8:401–413, 2012.

19 Michael Dinitz, Ama Koranteng, and Guy Kortsarz. Relative Survivable Network Design. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*, volume 245 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.41.

20 Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 750–759, 1999.

21 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.

22 András Frank. Kernel systems of directed graphs. *Acta Sci. Math.(Szeged)*, 41(1-2):63–76, 1979.

23 András Frank. *Connections in combinatorial optimization*, volume 38. Oxford University Press Oxford, 2011.

24 Naveen Garg, Goran Konjevod, and Ramamoorthi Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000. Preliminary version in Proc. of ACM-SIAM SODA 1998.

25 M. X. Goemans, A. V. Goldberg, S. Plotkin, D. B. Shmoys, E. Tardos, and D. P. Williamson. Improved approximation algorithms for network design problems. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 223–232, 1994.

26 M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In *Approximation algorithms for NP-hard problems*, pages 144–191. PWS Publishing Company, Boston, MA, 1997.

27 Fabrizio Grandoni, Bundit Laekhanukit, and Shi Li. $O(\log^2 k \log\log k)$-approximation algorithm for directed steiner tree: a tight quasi-polynomial-time algorithm. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 253–264, 2019.

28 A. Gupta and J. Könemann. Approximation algorithms for network design: A survey. *Surveys in Operations Research and Management Science*, 16(1):3–20, 2011.

29 Anupam Gupta, Ravishankar Krishnaswamy, and Ramamoorthi Ravi. Tree embeddings for two-edge-connected network design. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1521–1538. SIAM, 2010.

30 Eran Halperin, Guy Kortsarz, Robert Krauthgamer, Aravind Srinivasan, and Nan Wang. Integrality ratio for group steiner trees and directed steiner trees. *SIAM Journal on Computing*, 36(5):1494–1511, 2007.

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

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

33 Kamal Jain, Ion Măndoiu, Vijay V Vazirani, and David P Williamson. A primal-dual schema based approximation algorithm for the element connectivity problem. *Journal of Algorithms*, 45(1):1–15, 2002.

34 David R Karger. Global min-cuts in rnc, and other ramifications of a simple min-cut algorithm. In *Soda*, volume 93, pages 21–30, 1993.

35 David R Karger. Minimum cuts in near-linear time. *Journal of the ACM (JACM)*, 47(1):46–76, 2000.

36 G. Kortsarz and Z. Nutov. Approximating minimum cost connectivity problems. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2010.

37 Z. Nutov. Approximating Steiner networks with node-weights. *SIAM Journal on Computing*, 39(7):3001–3022, 2010.

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

39 Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 255–264, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1374376.1374415.

40 Gabriele Reich and Peter Widmayer. Beyond steiner's problem: A vlsi oriented generalization. In *International Workshop on Graph-theoretic Concepts in Computer Science*, pages 196–210. Springer, 1989.

41 D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15(3):435–454, 1995.