APPROXIMATION ALGORITHMS FOR FLEXIBLE GRAPH CONNECTIVITY

SYLVIA BOYD, JOSEPH CHERIYAN, ARASH HADDADAN, AND SHARAT IBRAHIMPUR

Abstract. We present approximation algorithms for several network design problems in the model of Flexible Graph Connectivity (Adjiashvili, Hommelsheim and Mühlenthaler, "Flexible Graph Connectivity", *Math. Program.* pp. 1–33 (2021), *IPCO* 2020: pp. 13–26).

Let $k \geq 1$, $p \geq 1$ and $q \geq 0$ be integers. In an instance of the (p, q)-Flexible Graph Connectivity problem, denoted (p, q)-FGC, we have an undirected connected graph $G = (V, E)$, a partition of E into a set of safe edges $\mathcal{S}$ and a set of unsafe edges $\mathcal{U}$, and nonnegative costs $c : E \to \mathbb{R}_{\geq 0}$ on the edges. A subset $F \subseteq E$ of edges is feasible for the (p, q)-FGC problem if for any set $F' \subseteq \mathcal{U}$ with $|F'| \leq q$, the subgraph $(V, F \setminus F')$ is p-edge connected. The algorithmic goal is to find a feasible solution F that minimizes $c(F) = \sum_{e \in F} c_e$. We present a simple 2-approximation algorithm for the $(1, 1)$-FGC problem via a reduction to the minimum-cost rooted 2-arborescence problem. This improves on the 2.527-approximation algorithm of Adjiashvili et al. Our 2-approximation algorithm for the $(1, 1)$-FGC problem extends to a $(k + 1)$-approximation algorithm for the $(1, k)$-FGC problem. We present a 4-approximation algorithm for the $(k, 1)$-FGC problem, and an $O(q \log |V|)$-approximation algorithm for the (p, q)-FGC problem. Finally, we improve on the result of Adjiashvili et al. for the unweighted $(1, 1)$-FGC problem by presenting a 16/11-approximation algorithm.

The (p, q)-FGC problem is related to the well-known *Capacitated k-Connected Subgraph* problem (denoted Cap-k-ECSS) that arises in the area of Capacitated Network Design. We give a $\min(k, 2u_{\max})$-approximation algorithm for the Cap-k-ECSS problem, where $u_{\max}$ denotes the maximum capacity of an edge.

Corresponding Author: Joseph Cheriyan
Affiliation: Department of Combinatorics and Optimization, University of Waterloo, Canada
E-mail Address: jcheriyan@uwaterloo.ca

Date: May 7, 2023.

1991 *Mathematics Subject Classification.* Primary 68W25; Secondary 90C17, 90C27, 90C59, 05C40.

Key words and phrases. Approximation Algorithms, Combinatorial Optimization, Network Design, Edge-Connectivity of Graphs, Reliability of Networks.

A preliminary version of this paper appeared in the Proceedings of the 41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021), December 15–17, 2021, Ed. M. Bojańczyk and C. Chekuri (LIPIcs, Volume 213, Article No. 9, pp. 9:1–9:14).

The second author is supported in part by NSERC, RGPIN-2019-04197.

This work of the third author was mostly done as a postdoctoral researcher at the Biocomplexity Institute and Initiative at the University of Virginia, Charlottesville, and supported by the NSF Expeditions in Computing Grant with award number CCF-1918656.

The fourth author is supported in part by NSERC grant 327620-09.

1. Introduction

Network design and graph connectivity are core topics in Theoretical Computer Science and Operations Research. A basic problem in network design is to find a minimum-cost sub-network H of a given network G such that H satisfies some specified connectivity requirements. Most of these problems are NP-hard. Several important algorithmic paradigms were developed in the context of these topics, ranging from exact algorithms for the shortest s,t-path problem and the minimum spanning tree (MST) problem to linear programming-based approximation algorithms for the survivable network design problem and the generalized Steiner network problem. Network design problems are often motivated by practical considerations such as the design of fault-tolerant supply chains, congestion control for urban road traffic, and the modeling of epidemics (see [15, 17, 20]).

Recently, Adjiashvili, Hommelsheim and Mühlenthaler [1, 2] introduced a new model called *Flexible Graph Connectivity* (FGC), that is motivated by research in robust optimization. (In this paper, the notation FGC may be used as an abbreviation for "the FGC problem"; we use similar abbreviations for the names of other related problems.) In an instance of FGC, we have an undirected connected graph $G = (V, E)$, a partition of E into a set of safe edges $\mathcal{S}$ and a set of unsafe edges $\mathcal{U}$, and nonnegative costs $c : E \to \mathbb{R}_{\geq 0}$ on the edges. The graph G may have multiedges, but no self-loops. We use n to denote the number of vertices of G. The cost of an edge-set $F \subseteq E$ is denoted by $c(F) = \sum_{e \in F} c_e$. A subset $F \subseteq E$ of edges is feasible for FGC if for any unsafe edge $e \in F \cap \mathcal{U}$, the subgraph $(V, F \setminus \{e\})$ is connected. The problem is to find a feasible edge-set F of minimum cost. The motivation for studying FGC is two-fold. First, FGC generalizes many well-studied survivable network design problems. Notably, the problem of finding a minimum-cost 2-edge connected spanning subgraph (abbreviated as 2-ECSS) corresponds to the special case of FGC where all edges are unsafe, and the MST problem corresponds to the special case of FGC where all edges are safe. Second, FGC captures a non-uniform model of survivable network design problems where a specified subset of edges never fail, whereas each edge can fail in the classical model of survivable network design problems. Since FGC generalizes the minimum-cost 2-ECSS problem, it is APX-hard (see [8]); thus, a polynomial-time approximation scheme for FGC is ruled out unless P=NP.

The notion of (p, q)-FGC is an extension of the basic FGC model where we have two additional integer parameters p and q satisfying $p \geq 1$ and $q \geq 0$. For a subgraph $H = (V, F)$ of G and a vertex-set $S \subseteq V$, we use $\delta_H(S)$ to denote the set of edges in H with exactly one end-vertex in S. A subset $F \subseteq E$ of edges is feasible for (p, q)-FGC if the spanning subgraph $H = (V, F)$ is p-edge connected, and moreover, the deletion of any set of at most q unsafe edges of F preserves p-edge connectivity. In other words, each cut $\delta_H(S)$, $\emptyset \neq S \subsetneq V$, of H either contains p safe edges or contains $p + q$ (safe or unsafe) edges. The algorithmic goal is to find a feasible edge-set F of minimum cost. The (p, q)-FGC problem is a natural and fundamental question in robust network design. See the appendix, Section 7, for an example of $(2, 2)$-FGC. Note that the FGC problem is the same as the $(1, 1)$-FGC problem.

Observe that the problem of finding a minimum-cost p-edge connected spanning subgraph (abbreviated as p-ECSS) corresponds to the special case of (p, q)-FGC where all edges are safe, and the problem of finding a minimum-cost $(p + q)$-ECSS corresponds to the special case of (p, q)-FGC where all edges are unsafe. Informally speaking, the model of (p, q)-FGC interpolates between p-edge connectivity (when all edges are safe) and $(p + q)$-edge connectivity (when all edges are unsafe).

For each of the problems considered in this paper, any solution (i.e., output) must be a subgraph of the graph G of the instance; that is, the (multi) set of edges F of a solution must be a subset of the (multi) set $E(G)$; in other words, the number of copies of an edge $e = vw$ in F cannot exceed the number of copies of e in $E(G)$; see the discussion in [18, Chapter 3.1].

The (p, q)-FGC model is related to the model of Capacitated Network Design. There are several results pertaining to approximation algorithms for various problems in Capacitated Network Design; for example, see Goemans et al. [9] and Chakrabarty et al. [3]. Let k be a positive integer. The Capacitated k-Connected Subgraph problem, see [3], is a well studied problem in this area. We denote this problem by Cap-k-ECSS. In an instance of this problem, we have an undirected connected graph $G = (V, E)$, nonnegative integer edge-capacities $u : E \to \mathbb{Z}_{\geq 0}$, nonnegative edge-costs $c : E \to \mathbb{R}_{\geq 0}$, and a positive integer k. The goal is to find an edge-set $F \subseteq E$ such that for any nonempty $R \subsetneq V$ we have $\sum_{e \in \delta_G(R) \cap F} u_e \geq k$, and $c(F)$ is minimized. Let n and m denote the number of vertices and edges of G, respectively. See the appendix, Section 7, for an example of Cap-4-ECSS. For this problem, Goemans et al. [9] give a $\min(2k, m)$-approximation algorithm, and Chakrabarty et al. [3] give a randomized $O(\log n)$-approximation algorithm. We mention that for some particular values of p and q, such as $p = 1$ (and arbitrary q) or $q \leq 1$ (and arbitrary p), (p, q)-FGC can be cast as a special case of the Cap-k-ECSS problem. The FGC problem corresponds to the Cap-2-ECSS problem where safe edges have capacity two and unsafe edges have capacity one. Moreover, $(1, k)$-FGC corresponds to the Cap-$(k + 1)$-ECSS problem where safe edges have capacity $k + 1$ and unsafe edges have capacity one; $(k, 1)$-FGC corresponds to the Cap-$(k(k+1))$-ECSS problem where safe edges have capacity $k+1$ and unsafe edges have capacity k. We mention that there exist values of p and q (e.g., $p = 2, q = 2$) such that the (p, q)-FGC problem differs from the Cap-$(p(p + q))$-ECSS problem where safe edges have capacity $p + q$ and unsafe edges have capacity p.

Our contributions: We list our main contributions and give a brief overview of our results and techniques.

Our first result is based on a simple reduction from FGC to the well-known minimum-cost rooted 2-arborescence problem that achieves an approximation factor of two for FGC. This result matches the current best approximation factor known for the minimum-cost 2-ECSS problem, and improves on the 2.527-approximation algorithm of [2]. At a high level, our result is based on an extension of the 2-approximation algorithm of Khuller and Vishkin [14] for the minimum-cost 2-ECSS problem. (In fact, Khuller and Vishkin [14] give a reduction from the minimum-cost k-ECSS problem to the problem of computing a minimum-cost

rooted k-arborescence in a digraph, and they prove an approximation factor of two for the former problem.)

Theorem 1.1. *There is a* 2*-approximation algorithm for* FGC*.*

The following result generalizes Theorem 1.1 to the $(1,k)$-FGC problem. Our proof of the generalization of Theorem 1.1 is based on a reduction from $(1,k)$-FGC to the minimum-cost rooted $(k+1)$-arborescence problem (see Definition 2.2 and Proposition 2.3). We lose a factor of $k+1$ in this reduction.

Theorem 1.2. *There is a* $(k+1)$*-approximation algorithm for* $(1,k)$-FGC*.*

In Section 3, we consider the Capacitated k-Connected Subgraph problem that we denote by Cap-k-ECSS. For notational convenience, let $u_{\max} := \max\{u_e : e \in E\}$ denote the maximum capacity of an edge in the given instance of Cap-k-ECSS; similarly, let $u_{\min} := \min\{u_e : e \in E\}$. Our main result in Section 3 is a $\min(k, 2u_{\max})$-approximation algorithm for the Cap-k-ECSS problem, stated in Theorem 1.3. Similar to Theorems 1.1 and 1.2, our proof of Theorem 1.3 is based on a reduction from the Cap-k-ECSS problem to the minimum-cost rooted k-arborescence problem. The factor m in the $\min(2k, m)$ approximation factor of Goemans et al. comes from the fact that a simple greedy strategy yields an m-approximation for the Cap-k-ECSS problem. Thus, we may assume that $k \leq m$ holds. Furthermore, if $2u_{\max} \leq m$, our result is better, and, in fact, for the standard case of $u_{\min} = 1,\ u_{\max} = k \ll m$, no previous result achieves an approximation factor of k (to the best of our knowledge). Our result above is incomparable to the result in [3]; our approximation factor is independent of the graph size, whereas their result is independent of k. The algorithm in [3] is probabilistic and its analysis is based on Chernoff tail bounds.

Theorem 1.3. *There is an* α*-approximation algorithm for the* Cap-k-ECSS *problem, where* $\alpha := \min(k, 2u_{\max})$*.*

In Section 4, we present a 4-approximation algorithm for the $(k,1)$-FGC problem, see Theorem 1.4. Our algorithm in Theorem 1.4 runs in two stages. In the first stage we pretend that all edges are safe. Under this assumption, $(k,1)$-FGC simplifies to the minimum-cost k-ECSS problem, for which several 2-approximation algorithms are known, see [14], [12]. We apply one of these algorithms. Let $H = (V, F)$ be the k-ECSS found in Stage 1. In the second stage, our goal is to preserve k-edge connectivity against the failure of any one unsafe edge. In the graph H, consider a cut $\delta_H(S)$, $\emptyset \neq S \subsetneq V$, that has (exactly) k edges and that contains at least one unsafe edge. Such a cut, that we call deficient, certifies that F is not feasible for $(k,1)$-FGC, so it needs to be augmented. The residual problem is that of finding a cheapest augmentation of all deficient cuts w.r.t. (with respect to) F. It turns out that this augmentation problem can be formulated as the minimum-cost f-connectivity problem for an uncrossable function f (to be defined in Section 4). Williamson, Goemans, Mihail and Vazirani [22] present a 2-approximation algorithm for the latter problem.

Theorem 1.4. *There is a* 4*-approximation algorithm for* $(k,1)$-FGC*.*

In Section 5, we present an $O(q \log n)$-approximation algorithm for (p, q)-FGC. As above, our approximation algorithm for (p, q)-FGC runs in two stages. In the first stage, we construct an instance of the Cap-k-ECSS problem (that partially models the given (p, q)-FGC instance), and then we apply our approximation algorithm for the Cap-k-ECSS problem (see Theorem 1.3) to compute a cheap edge-set F that is nearly feasible for the (p, q)-FGC instance. We call a cut $\delta_G(S)$, $\emptyset \neq S \subsetneq V$, deficient (w.r.t. F) if $|F \cap \delta_G(S) \cap \mathcal{S}| < p$ and $|F \cap \delta_G(S)| < p + q$; thus, a deficient cut is one that certifies the infeasibility of F. The second stage of our algorithm applies several iterations. In the first iteration, we find all the deficient cuts of our current subgraph $H = (V, F)$, and then we apply the greedy algorithm for the well-known hitting-set problem to cover all the deficient cuts. We repeat such iterations until we attain feasibility in the given (p, q)-FGC instance (i.e., there are no deficient cuts).

Theorem 1.5. *There is an $O(q \log n)$-approximation algorithm for (p, q)-FGC.*

In Section 6, we consider the unweighted version of FGC, where each edge has unit cost. We design an improved approximation algorithm for this special case. We give two algorithms for obtaining two candidate solutions to an instance of unweighted FGC; the simpler of these algorithms is discussed by Adjiashvili et al. [1, 2]. Assuming that we have access to an α-approximation algorithm for the minimum-size (i.e., unweighted) 2-ECSS problem, we argue that the cheaper of the two candidate solutions is a $\frac{4\alpha}{2\alpha+1}$-approximate solution to the instance of unweighted FGC. We can take $\alpha = 4/3$ by using the result of Sebö and Vygen [19] or Hunkenschröder, Vempala and Vetta [11].

Theorem 1.6. *There is a $\frac{16}{11}$-approximation algorithm for unweighted* FGC.

Section 2 focuses on $(1, k)$-FGC and gives our $(k + 1)$-approximation for this problem (Theorem 1.2); the 2-approximation for FGC (Theorem 1.1) follows as a special case. Section 3 focuses on the Cap-k-ECSS problem, and gives our $\min(k, 2u_{\max})$-approximation algorithm for this problem (Theorem 1.3). Section 4 focuses on $(k, 1)$-FGC, and gives our 4-approximation algorithm for this problem (Theorem 1.4). Section 5 focuses on (p, q)-FGC, and gives our $O(q \log n)$-approximation algorithm for this problem (Theorem 1.5). Section 6 focuses on the unweighted version of FGC, and gives our $\frac{16}{11}$-approximation algorithm for this problem (Theorem 1.6). This section also has an improved approximation factor for the unweighted version of $(k, 1)$-FGC. The appendix, Section 7, presents an example of $(2, 2)$-FGC and an example of Cap-4-ECSS.

2. A $(k + 1)$-Approximation Algorithm for $(1, k)$-FGC

We give a $(k + 1)$-approximation for $(1, k)$-FGC, where k is a positive integer. The 2-approximation for FGC (Theorem 1.1) follows as a special case. Recall that in an instance of $(1, k)$-FGC we have an undirected graph $G = (V, E)$ (with no self loops), a partition of E into safe and unsafe edges, $E = \mathcal{S} \dot{\cup} \mathcal{U}$, and nonnegative edge-costs $c : E \to \mathbb{R}_{\geq 0}$. Our objective is to find a minimum-cost edge-set $F \subseteq E$

such that the subgraph (V, F) remains connected against the failure of any set of k unsafe edges.

For a subgraph H of G and a vertex-set $S \subseteq V$, we use $\delta_H(S)$ or $\delta_{E(H)}(S)$ to denote the set of edges in H with exactly one end-vertex in S, i.e., $\delta_H(S) := \{e = uv \in E(H) : |\{u, v\} \cap S| = 1\}$. We drop the subscript when the underlying graph is clear from the context.

The following characterization of feasible solutions of $(1, k)$-FGC is straightforward.

Proposition 2.1. *An edge-set $F \subseteq E$ is feasible for $(1, k)$-FGC if and only if for all nonempty $S \subsetneq V$, the edge-set $F \cap \delta(S)$ contains a safe edge or $k + 1$ unsafe edges. Furthermore, in time polynomial in n, we can test if F is feasible for $(1, k)$-FGC.*

We check the feasibility of F for $(1, k)$-FGC by creating an auxiliary capacitated graph that has a capacity of $k + 1$ for each safe edge and a capacity of one for each unsafe edge; then, we test whether or not the minimum capacity of a cut of the auxiliary graph is at least $k + 1$. For the rest of this section, we assume that the given instance of $(1, k)$-FGC is feasible.

As mentioned before, our algorithm for $(1, k)$-FGC is based on a reduction to the minimum-cost rooted $(k + 1)$-arborescence problem. We state a few standard results on arborescences. Let $D = (W, A)$ be a digraph and let $c' : A \to \mathbb{R}_{\geq 0}$ be nonnegative costs on the arcs. We remark that D may have parallel arcs but it has no self-loops. Let $r \in W$ be a designated root vertex. For a subgraph H of D and a nonempty vertex-set $S \subsetneq W$, we use $\delta^{\text{in}}_H(S)$ to denote the set of arcs in H such that the head of the arc is in S and the tail of the arc is in $W \setminus S$, i.e., $\delta^{\text{in}}_H(S) := \{a = (u, v) \in A(H) : u \notin S, v \in S\}$.

Definition 2.1 (r-rooted arborescence)**.** *An r-rooted arborescence (W, T) is a subgraph of D satisfying: (i) the undirected version of T is acyclic; and (ii) for every $v \in W \setminus \{r\}$, there is a directed path from r to v in the subgraph (W, T).*

In other words, an r-rooted arborescence is a directed spanning tree such that vertex r has no incoming arcs and every other vertex has one incoming arc. An r-rooted k-arborescence is a union of k arc-disjoint r-rooted arborescences.

Definition 2.2 (r-rooted k-arborescence)**.** *For a positive integer k, a subgraph (W, T) is an r-rooted k-arborescence if T can be partitioned into k arc-disjoint r-rooted arborescences.*

The following results on rooted arborescences and the corresponding optimization problem are useful for us.

Proposition 2.2 ([18], Chapter 53.8)**.** *Let $D = (W, A)$ be a digraph, let $r \in W$ be a vertex, and let k be a positive integer. Then, D contains an r-rooted k-arborescence if and only if $|\delta^{\text{in}}_D(S)| \geq k$ for any nonempty vertex-set $S \subseteq V \setminus \{r\}$.*

Proposition 2.3 ([18], Theorem 53.10)**.** *In strongly polynomial time, we can obtain an optimal solution to the minimum-cost r-rooted k-arborescence problem on (D, c'), or conclude that there is no r-rooted k-arborescence in D.*

Claim 2.4. *Let (W, T) be an r-rooted k-arborescence for an integer $k \geq 1$. Let $u, v \in W$ be two distinct vertices. Then, the number of arcs in T that have one end-vertex at u and the other end-vertex at v (counting multiplicities) is at most k.*

Proof. Since an r-rooted k-arborescence is a union of k arc-disjoint r-rooted 1-arborescences, it suffices to prove the result for $k = 1$. The claim holds for $k = 1$ because the undirected version of T is acyclic, by definition. □

Informally speaking, our proofs map undirected graphs to their directed counterparts by bidirecting edges. We formalize this notion.

Definition 2.3 (Bidirected pair)**.** *For an undirected edge $e = uv$, we call the arc-set $\{(u, v), (v, u)\}$ a bidirected pair arising from e.*

The following lemma shows how a solution F to $(1, k)$-FGC can be used to obtain a rooted $(k + 1)$-arborescence (in an appropriate digraph) of cost at most $(k + 1)$ times $c(F)$.

Lemma 2.5. *Let F be a $(1, k)$-FGC solution. Consider the digraph $D = (V, A)$ where the arc-set A is defined as follows: for each unsafe edge $e \in F \cap \mathcal{U}$, we include a bidirected pair of arcs arising from e, and for each safe edge $e \in F \cap \mathcal{S}$, we include $k + 1$ bidirected pairs arising from e. Consider the natural extension of the cost vector c to D where the cost of an arc $(u, v) \in A$ is equal to the cost of the edge in G that gives rise to it. Then, there is an r-rooted $(k + 1)$-arborescence in D with cost at most $(k + 1)c(F)$.*

Proof. Let (V, T) be a minimum-cost r-rooted $(k + 1)$-arborescence in D. First, we argue that T is well-defined. By Proposition 2.2, it suffices to show that for any nonempty $S \subseteq V \setminus \{r\}$, we have $|\delta^{\text{in}}_D(S)| \geq k + 1$. Fix some nonempty $S \subseteq V \setminus \{r\}$. By feasibility of F, $F \cap \delta(S)$ contains a safe edge or $k+1$ unsafe edges (see Proposition 2.1). If $F \cap \delta(S)$ contains a safe edge $e = uv$ with $v \in S$, then by our choice of A, $\delta^{\text{in}}_D(S)$ contains $k + 1$ (u, v)-arcs. Otherwise, $F \cap \delta(S)$ contains $k + 1$ unsafe edges, and for each such unsafe edge uv with $v \in S$, $\delta^{\text{in}}_D(S)$ contains the arc (u, v). In both cases we have $|\delta^{\text{in}}_D(S)| \geq k + 1$, so T is well-defined.

We use Claim 2.4 to show that T satisfies the required bound on the cost. For each unsafe edge $e \in F$, T contains at most 2 arcs from the bidirected pair arising from e, and for each safe edge $e \in F$, T contains at most $k+1$ arcs from the (disjoint) union of $k+1$ bidirected pairs arising from e. Thus, $c(T) \leq 2\, c(F \cap \mathcal{U}) + (k+1)\, c(F \cap \mathcal{S}) \leq (k + 1)\, c(F)$. □

Lemma 2.5 naturally suggests a reduction from $(1, k)$-FGC to the minimum-cost r-rooted $(k+1)$-arborescence problem. We prove the main theorem of this section.

Proof of Theorem 1.2. Fix some vertex $r \in V$ as the root vertex. Consider the digraph $D = (V, A)$ obtained from G as follows: for each unsafe edge $e \in \mathcal{U}$, we include a bidirected pair arising from e, and for each safe edge $e \in \mathcal{S}$, we include $k + 1$ bidirected pairs arising from e. For each edge $e \in E$, let $R(e)$ denote the multi-set of all arcs in D that arise from $e \in E$. For any edge $e \in E$ (that could be one of the copies of a multiedge) and each of the corresponding arcs $\vec{e} \in R(e)$, we

define $c_{\vec{e}} := c_e$. Let (V, T) denote a minimum-cost r-rooted $(k+1)$-arborescence in (D, c); note that the size of D is polynomial in the size of G since we assume that $k \leq |E|$, see the paragraph preceding Theorem 1.3 in Section 1. By Lemma 2.5, $c(T) \leq (k+1)c(F^*)$, where F^* denotes an optimal solution to the given instance of $(1,k)$-FGC.

We finish the proof by arguing that T induces a $(1,k)$-FGC solution F with cost at most $c(T)$. Let $F := \{e \in E : R(e) \cap T \neq \emptyset\}$. By definition of F and our choice of arc-costs in D, we have $c(F) \leq c(T)$. It remains to show that F is feasible for $(1,k)$-FGC. Consider a nonempty set $S \subseteq V \setminus \{r\}$. Since T is an r-rooted $(k+1)$-arborescence, by Proposition 2.2 we have $|\delta^{\text{in}}_T(S)| \geq k+1$. If $\delta^{\text{in}}_T(S)$ contains a safe arc (i.e., an arc that arises from a safe edge), then that safe edge belongs to $F \cap \delta(S)$. Otherwise, $\delta^{\text{in}}_T(S)$ contains some $k+1$ unsafe arcs (that arise from unsafe edges). Since the two orientations of an edge cannot appear simultaneously in $\delta^{\text{in}}_D(S)$, we have $|F \cap \mathcal{U} \cap \delta(S)| \geq k+1$. By Proposition 2.1, F is a feasible solution for the given instance of $(1,k)$-FGC with $c(F) \leq (k+1) \cdot \text{OPT}$, where OPT denotes the optimal value of the instance. □

3. The Capacitated k-Connected Subgraph Problem

In this section we consider the Cap-k-ECSS problem. We are given a graph $G = (V, E)$, nonnegative integer edge-capacities $u : E \to \mathbb{Z}_{\geq 0}$, nonnegative edge-costs $c : E \to \mathbb{R}_{\geq 0}$, and a positive integer k. Our goal is to find a spanning subgraph $H = (V, F)$ such that for all nonempty sets $R \subsetneq V$ we have $u(\delta_F(R)) \geq k$, and the cost $c(F)$ is minimized.

Given an instance of the Cap-k-ECSS problem, we may assume without loss of generality that $u_e \in \{1, \ldots, k\}$ for all $e \in E$ (we can drop edges with zero capacity and replace edge-capacities $\geq k+1$ by k). We also assume that the Cap-k-ECSS instance is feasible. This can be verified in polynomial time by using a min-cut algorithm: the instance is infeasible if and only if G has a cut $\delta(S)$, $\emptyset \neq S \subsetneq V$, with capacity $u(\delta(S))$ strictly less than k. Let $u_{\max} = \max_{e \in E} u_e$ denote the maximum capacity of an edge in G. Our main result in this section is a $\min(k, 2u_{\max})$-approximation algorithm for the Cap-k-ECSS problem (Theorem 1.3); our algorithm is based on a reduction to the minimum-cost rooted k-arborescence problem.

Description of our algorithm for the Cap-k-ECSS **problem:** Let $D = (V, A)$ be the directed graph obtained from G by replacing every edge $xy \in E$ by u_{xy} pairs of bidirected arcs $(x, y), (y, x)$, each with the same cost as the edge xy; thus, each edge e in G has $2u_e$ corresponding arcs in D, each of cost c_e. Designate an arbitrary vertex $r \in V$ as the root. By feasibility of the Cap-k-ECSS instance, we know that D contains an r-rooted k-arborescence (see Proposition 2.2). We use Proposition 2.3 on (D, c) to obtain a minimum-cost r-rooted k-arborescence (V, T') in polynomial time. Let F' be the set of all edges $e \in E$ such that at least one of the $2u_e$ corresponding arcs in D appears in T'.

Lemma 3.1. *The edge-set F' obtained by the above algorithm is feasible for the given* Cap-k-ECSS *instance and it has cost at most $c(T')$.*

Proof. Let $R \subsetneq V \setminus \{r\}$ be a nonempty vertex-set that excludes the root vertex r. Since (V, T') contains k arc-disjoint r-rooted arborescences, $|\delta^{\text{in}}_{T'}(R)| \geq k$ (by Proposition 2.2). For each edge $e \in E$, at most u_e of the corresponding arcs in D can occur in the arc-set $\delta^{\text{in}}_{T'}(R)$, by the construction of D; hence, for any edge $e \in \delta(R) \cap F'$, u_e is an upper bound on the number of corresponding arcs of e in $\delta^{\text{in}}_{T'}(R)$. Therefore, $\sum_{e \in \delta(R) \cap F'} u_e \geq |\delta^{\text{in}}_{T'}(R)| \geq k$, and F' is a feasible solution for the Cap-k-ECSS instance, as required. For any edge $e \in E$, we only include a single copy of e in F' whenever any of the $2u_e$ corresponding arcs appear in T', so we have $c(F') \leq c(T')$. □

We now prove Theorem 1.3 by showing that the edge-set F' found by the above algorithm has cost at most $\min(k, 2u_{\max}) \cdot \text{OPT}$, where OPT denotes the optimal value of the instance.

Proof of Theorem 1.3. Let $(G(V, E), \{c_e\}_{e \in E}, \{u_e\}_{e \in E}, k)$ denote a feasible instance of the Cap-k-ECSS problem. Let $r \in V$ be the root vertex fixed by the above algorithm. Let $D = (V, A)$ be the digraph and let (V, T') be the r-rooted k-arborescence constructed by the algorithm.

Let F^* be an optimal solution to the Cap-k-ECSS instance, and let $D^* = (V, A^*)$ be the digraph obtained from (V, F^*) by replacing every edge $xy \in F^*$ by u_{xy} pairs of bidirected arcs $(x, y), (y, x)$ each with the same cost as edge xy. By feasibility of F^* (for the Cap-k-ECSS instance), D^* contains a k-arborescence rooted at r. Let (V, T^*) denote an optimal r-rooted k-arborescence in D^*.

Since D^* is a subgraph of D and (V, T') is an optimal r-rooted k-arborescence in D, we have $c(T') \leq c(T^*)$. By Lemma 3.1, we can prove the theorem by arguing that $c(T^*) \leq \min(k, 2u_{\max}) c(F^*)$. We prove this inequality by examining two cases.

Case 1: Observe that for any edge $e \in F^*$ there are at most $2u_e$ corresponding arcs in A^* by construction of D^*. Hence, we have $c(T^*) \leq c(A^*) \leq 2u_{\max} c(F^*)$.

Case 2: Next, observe that T^* can be partitioned into k (arc-disjoint) r-rooted arborescences, each of which can use at most one of the $2u_e$ corresponding arcs of an edge e of G; see Claim 2.4. It follows that for each edge $e \in F^*$ at most k of the $2u_e$ corresponding arcs can appear in T^*. Hence, $c(T^*) \leq k\, c(F^*)$.

This completes the proof. □

4. A 4-Approximation Algorithm for $(k, 1)$-FGC

Our main result in this section is a 4-approximation algorithm for $(k, 1)$-FGC (Theorem 1.4). Recall that in an instance of $(k, 1)$-FGC, we have a graph $G = (V, E)$, with a partition of the edge-set into safe and unsafe edges, $E = \mathcal{S} \cup \mathcal{U}$, nonnegative edge-costs $c : E \to \mathbb{R}_{\geq 0}$, and a positive integer k. The objective is to find a minimum-cost subgraph that remains k-edge connected against the failure of any one unsafe edge. We remark that for the $k = 1$ case, Theorem 1.1 yields a better approximation factor than Theorem 1.4. Let F^* denote an optimal solution

to the $(k,1)$-FGC instance. The following result pertains to feasible solutions of $(k,1)$-FGC.

Proposition 4.1. *An edge-set $F \subseteq E$ is feasible for $(k,1)$-FGC if and only if for all nonempty $S \subsetneq V$, the edge-set $F \cap \delta(S)$ contains k safe edges or $k+1$ edges. Furthermore, in time polynomial in n, we can test if F is feasible for $(k,1)$-FGC.*

Proof. The characterization of feasible solutions of $(k,1)$-FGC follows from the definitions.

We check the feasibility of F for $(k,1)$-FGC by creating an auxiliary capacitated graph that has a capacity of $k+1$ for each safe edge and a capacity of k for each unsafe edge; then, we test whether or not the minimum capacity of a cut of the auxiliary graph is at least $k(k+1)$. □

For the rest of this section, we assume that the given instance of $(k,1)$-FGC is feasible.

Proposition 4.1 suggests a two-stage strategy for finding an approximately optimal solution to $(k,1)$-FGC. In the first stage, we do not distinguish between safe edges and unsafe edges, and we compute a cheap k-ECSS of $G=(V,E)$ that we denote by $H_1=(V,F_1)$. Clearly, for every nonempty set $S \subsetneq V$, we have $|\delta_{H_1}(S)| \geq k$; if equality holds, then we call $\delta_{H_1}(S)$ a *k-edge-cut*. If F_1 is feasible for $(k,1)$-FGC, then we are done. Otherwise, by Proposition 4.1, the infeasibility of F_1 for $(k,1)$-FGC is due to k-edge-cuts of H_1 that contain at least one unsafe edge. We call such cuts *deficient*. In the second stage, we address the remaining augmentation problem for the deficient cuts, by casting it as a special case of the minimum-cost f-connectivity problem (defined below).

An instance of the minimum-cost f-connectivity problem consists of an undirected graph $G'=(V',E')$, nonnegative edge-costs $c' : E' \to \mathbb{R}_{\geq 0}$, and a requirement function $f : 2^{V'} \to \{0,1\}$ satisfying $f(\emptyset)=f(V)=0$. We assume access to f via a value oracle that takes as input a vertex-set $S \subseteq V$ and outputs $f(S)$. An edge-set $F \subseteq E'$ is feasible if $|F \cap \delta_{G'}(S)| \geq f(S)$ for every $S \subseteq V'$. In other words, F is feasible if and only if for every vertex-set S with $f(S)=1$ there is at least one F-edge in the cut $\delta(S)$. The objective is to find a feasible $F \subseteq E'$ that minimizes $c'(F)$. The minimum-cost f-connectivity problem can be formulated as an integer program whose linear relaxation (P) is stated below. For each edge $e \in E'$, the LP (linear program) has a nonnegative variable x_e; informally speaking, x_e quantifies the "usage" of the edge e in a feasible solution to the LP.

$$
\begin{aligned}
\text{(P)} \qquad & \min \sum_{e \in E'} c'_e x_e & & \\
& \text{subject to } x(\delta_{G'}(S)) \geq f(S) & & \forall\, S \subseteq V' \\
& x_e \geq 0 & & \forall\, e \in E'.
\end{aligned}
$$

The minimum-cost f-connectivity problem has received attention since it captures many well-known problems in network design. In particular, it captures the generalized Steiner network problem. Williamson et al. [22] gave a primal-dual

framework to obtain approximation algorithms for the minimum-cost f-connectivity problem when f is a proper function, and more generally, when f is an uncrossable function (also see the book chapter by Goemans and Williamson [10]).

Definition 4.1 (Uncrossable function)**.** *A function $f : 2^{V'} \to \{0,1\}$ is called uncrossable if $f(V') = 0$ and f satisfies the following two conditions:*

(i) f is symmetric, i.e., $f(S) = f(V' \setminus S)$ for all $S \subseteq V'$;

(ii) for any two sets $A, B \subseteq V'$ with $f(A) = f(B) = 1$, either $f(A \cap B) = f(A \cup B) = 1$ or $f(A \setminus B) = f(B \setminus A) = 1$ holds.

Under the assumption that *minimal violated sets* can be computed efficiently throughout, the primal-dual algorithm of [22] gives a 2-approximation for the minimum-cost f-connectivity problem with an uncrossable function f. There is no explicit result in [22] that can be quoted verbatim and applied for our purposes, so we state the most relevant result from [22].

Definition 4.2 (Minimal violated sets)**.** *Let $f : 2^{V'} \to \{0,1\}$ be a requirement function and $F \subseteq E'$ be an edge-set. A vertex-set $S \subseteq V'$ is said to be violated, w.r.t. f and F, if $f(S) = 1$ and $F \cap \delta_{G'}(S) = \emptyset$. We say that S is a minimal violated set if none of the proper subsets of S is violated.*

Proposition 4.2 ([22], Lemma 2.1)**.** *Let (G', c', f) be an instance of the minimum-cost f-connectivity problem, where $f : 2^{V'} \to \{0,1\}$ is an uncrossable function that is given via a value oracle. Suppose that for any $F \subseteq E'$ we can compute all minimal violated sets (w.r.t. f and F) in polynomial time. Then, in polynomial time, we can compute a feasible solution $F \subseteq E'$ such that $c'(F) \leq 2\, z^*$, where z^* denotes the optimal value of the LP relaxation* (P).

We now describe a two-stage algorithm that produces a 4-approximate $(k,1)$-FGC solution in polynomial time, thereby proving Theorem 1.4.

Description of our 4-approximation algorithm for $(k,1)$-FGC: Our algorithm runs in two stages. In the first stage, we construct an instance of the minimum-cost k-ECSS problem from the instance of $(k,1)$-FGC, by ignoring the distinction between the safe edges and the unsafe edges of G; the resulting instance is feasible because (V, F^*) is k-edge connected. Then, we compute a k-ECSS $H_1 = (V, F_1)$ of G by applying a 2-approximation algorithm to the instance of the minimum-cost k-ECSS problem; either the algorithm of Khuller & Vishkin [14] or the algorithm of Jain [12] could be used. Clearly, $c(F_1) \leq 2\, c(F^*)$. Next, we define the collection $\mathcal{C} := \{S \subseteq V : |\delta(S) \cap F_1| = k\}$ of all vertex-sets that correspond to k-edge-cuts of H_1. Consider the requirement function $f : 2^V \to \{0,1\}$ where

$$f(S) = 1 \text{ if and only if } S \in \mathcal{C} \text{ and } F_1 \cap \delta(S) \cap \mathcal{U} \neq \emptyset. \tag{4:1}$$

Consider an instance of the minimum-cost f-connectivity problem for the graph $G' := G - F_1$ with nonnegative edge-costs $c : (E \setminus F_1) \to \mathbb{R}_{\geq 0}$; note that $F^* \setminus F_1$ is feasible for this instance. In the second stage, we use Proposition 4.2 to compute a feasible solution $F_2 \subseteq E \setminus F_1$ for this instance such that $c(F_2) \leq 2\, c(F^* \setminus F_1)$. We return the solution $F = F_1 \dot{\cup} F_2$.

We prove Theorem 1.4 via a sequence of lemmas and claims. Lemma 4.3 below shows that f is uncrossable, and Claim 4.4 below shows that we can compute minimal violated sets (w.r.t. f and any $F' \subseteq E \setminus F_1$) in polynomial time. These two results together with Proposition 4.2 imply that our algorithm runs in polynomial time. Lemma 4.5 shows the correctness of our algorithm, and proves the approximation factor of four.

Lemma 4.3. *f is uncrossable.*

Proof. We check that the two properties (i), (ii) of an uncrossable function hold for f (recall Definition 4.1). The symmetry of f follows from the symmetry of cuts in undirected graphs. To check the second property, consider nonempty $A, B \subsetneq V$ satisfying $f(A) = f(B) = 1$. By definition of f, see (4:1), in the subgraph $H_1 = (V, F_1)$, both $\delta_{F_1}(A)$ and $\delta_{F_1}(B)$ are (minimum) k-edge-cuts, and there is at least one unsafe edge in each of these cuts. Let e_1 be an unsafe edge in $\delta_{H_1}(A)$ and let e_2 be an unsafe edge in $\delta_{H_1}(B)$. Let $r \in V$ be an arbitrary vertex. By symmetry of the cut function, we may assume without loss of generality that $r \notin A \cup B$. If $A \cap B = \emptyset$, then $f(A \setminus B) = f(B \setminus A) = 1$, so we are done. If $A \subseteq B$ or $A \supseteq B$, then $f(A \cap B) = f(A \cup B) = 1$, so we are done. Thus, we may assume that $A \cap B, V \setminus (A \cup B), A \setminus B, B \setminus A$ are all nonempty. For $S, T \subseteq V$, let $E(S, T)$ denote the set of edges of G with exactly one end-vertex in S and exactly one end-vertex in T. By submodularity of the function $d(S) := |\delta_{H_1}(S)|$, see [18], we have:

$$|\delta_{H_1}(A \cap B)| = |\delta_{H_1}(A \cup B)| = |\delta_{H_1}(A \setminus B)| = |\delta_{H_1}(B \setminus A)| = k. \tag{4:2}$$

Furthermore, we also have:

$$F_1 \cap E(A \setminus B, B \setminus A) = \emptyset \quad \text{and} \quad F_1 \cap E(A \cap B, V \setminus (A \cup B)) = \emptyset. \tag{4:3}$$

We finish the proof by a case analysis on e_1 and e_2. By (4:3), exactly one of the following cases occurs: (i) $e_1 \in E(A \setminus B, V \setminus (A \cup B))$; or (ii) $e_1 \in E(A \cap B, B \setminus A)$. If (i) occurs, then $f(A \setminus B) = f(A \cup B) = 1$. Otherwise, (ii) occurs and $f(A \cap B) = f(B \setminus A) = 1$. We apply a similar analysis for e_2. Exactly one of the following occurs: (a) $e_2 \in E(B \setminus A, V \setminus (A \cup B))$; or (b) $e_2 \in E(A \cap B, A \setminus B)$. If (a) occurs, then $f(B \setminus A) = f(A \cup B) = 1$. Otherwise, (b) occurs and $f(A \cap B) = f(A \setminus B) = 1$. It is easy to verify that for each of the four combinations, we either have $f(A \cap B) = f(A \cup B) = 1$ or we have $f(A \setminus B) = f(B \setminus A) = 1$. □

Claim 4.4. *For any $F' \subseteq E \setminus F_1$, we can compute all minimal violated sets w.r.t. f and F' in polynomial time.*

Proof. The number of minimum-cuts in a connected graph on n vertices is $O(n^2)$ (see [13]). Hence, we have $|\mathcal{C}| = O(|V|^2)$. Using results on network flow algorithms, we can compute $\mathcal{C}$ in polynomial time, see [4], [16]. Since we have explicit access to $\mathcal{C}$, we have a value oracle for f.

Let $F' \subseteq E \setminus F_1$ be a given edge-set. By Definition 4.2, any violated set S w.r.t. F' is in $\mathcal{C}$ and has $f(S) = 1$. We can exhaustively check each of the sets in $\mathcal{C}$ and find each of the minimal violated sets. □

Lemma 4.5. *The edge-set $F = F_1 \dot{\cup} F_2$ is feasible for $(k, 1)$-FGC and satisfies $c(F) \leq 4c(F^*)$.*

Proof. We first argue that F is feasible. Since F_1 and F_2 are edge-disjoint, we have $F \subseteq E$. We use the characterization of feasible solutions given by Proposition 4.1. Consider an arbitrary nonempty vertex-set $S \subsetneq V$. Since $H_1 = (V, F_1)$ is a k-edge connected subgraph of G, we have $|F_1 \cap \delta(S)| \geq k$. If $|F_1 \cap \delta(S)| \geq k+1$, then $|F \cap \delta(S)| \geq k+1$. Otherwise, $\delta(S)$ is a k-edge-cut of H_1, i.e., $S \in \mathcal{C}$. If $F_1 \cap \delta(S)$ contains only safe edges, then $F \cap \delta(S)$ contains k safe edges. Otherwise, by definition of f, see (4:1), $f(S) = 1$. Next, by feasibility of F_2 for the minimum-cost f-connectivity problem, we have $F_2 \cap \delta(S) \neq \emptyset$. Thus, $|F \cap \delta(S)| = |F_1 \cap \delta(S)| + |F_2 \cap \delta(S)| \geq k+1$. We show that $c(F) \leq 4\, c(F^*)$ by arguing that each of $c(F_1)$ and $c(F_2)$ is $\leq 2\, c(F^*)$. The bound on $c(F_1)$ is immediate from the fact that F^* is feasible for the instance of the minimum-cost k-ECSS problem considered in Stage 1, and by the 2-approximation algorithm used in Stage 1. Next, by feasibility of $F^* \setminus F_1$ for the instance of the minimum-cost f-connectivity problem, we have $c(F_2) \leq 2c(F^* \setminus F_1) \leq 2c(F^*)$. The lemma follows. □

Remarks: The function f is not a proper function (see [22]), and it is not weakly-supermodular (see [12]). Any proper function $g : 2^V \to \{0, 1\}$ must satisfy the maximality property, that is, $g(A \cup B) \leq \max\{g(A), g(B)\}$ must hold for any two disjoint sets $A, B \subseteq V$. Suppose that $k = 2$. Consider the graph $H_1 = (V, F_1)$ with $V = \{v_1, v_2, v_3\}$ and with four unsafe edges, namely, two copies of v_1v_2, one copy of v_1v_3, and one copy of v_2v_3. Let f be defined by (4:1). Then we have $f(\{v_1\}) = 0$, $f(\{v_2\}) = 0$, and $f(\{v_1, v_2\}) = 1$. Clearly, maximality is violated by the sets $A = \{v_1\}, B = \{v_2\}$. Any weakly-supermodular function $g : 2^V \to \mathbb{Z}$ must satisfy the property that the inequality $g(A) + g(B) \leq \max(g(A - B) + g(B - A),\ g(A \cap B) + g(A \cup B))$ holds for any two sets $A, B \subseteq V$. Suppose that $k = 2$. Consider the graph $H_1 = (V, F_1)$ with $V = \{v_1, v_2, v_3, v_4\}$ and with one unsafe edge, namely, v_2v_3, and five safe edges, namely, two copies of v_1v_2, two copies of v_3v_4, and one copy of v_4v_1. Let f be defined by (4:1). Let $A = \{v_1, v_2\}$ and let $B = \{v_2, v_3\}$. Then we have $f(A) = 1$, $f(B) = 0$, $f(A - B) = f(\{v_1\}) = 0$, $f(B - A) = f(\{v_3\}) = 0$, $f(A \cap B) = f(\{v_2\}) = 0$, and $f(A \cup B) = f(\{v_1, v_2, v_3\}) = 0$. Clearly, the required inequality fails to hold for the sets A, B.

5. An $O(q \log n)$-Approximation Algorithm for (p, q)-FGC

In this section, we present an $O(q \log n)$-approximation algorithm for (p, q)-FGC. Recall that an instance of (p, q)-FGC consists of an undirected graph $G = (V, E)$, a partition of E into safe and unsafe edges, $E = \mathcal{S} \dot\cup \mathcal{U}$, nonnegative edge-costs $c : E \to \mathbb{R}_{\geq 0}$, and two integer parameters $p \geq 1$ and $q \geq 0$. The objective is to find a minimum-cost edge-set $F \subseteq E$ such that the subgraph (V, F) remains p-edge connected against the failure of any set of at most q unsafe edges, that is, for any $F' \subseteq \mathcal{U}$ with $|F'| \leq q$, the subgraph $(V, F \setminus F')$ is p-edge connected. We assume that $q \geq 2$, since otherwise our results from Section 4 give a 4-approximation algorithm. The following result pertains to feasible solutions of (p, q)-FGC.

Proposition 5.1. *Consider an instance of* (p, q)-FGC. *An edge-set* $F \subseteq E$ *is feasible if and only if for all nonempty* $S \subsetneq V$, *the edge-set* $F \cap \delta(S)$ *contains* p

safe edges or $p+q$ edges. Furthermore, in time polynomial in n, we can test if F is feasible for (p,q)-FGC.

Proof. The characterization of feasible solutions of (p,q)-FGC follows from the definitions.

We check the feasibility of F for (p,q)-FGC by creating an auxiliary capacitated graph that has $G=(V,F)$ as the underlying graph, and that has a capacity of $p+q$ for each safe edge and a capacity of p for each unsafe edge. Then, we compute a minimum-capacity cut of the auxiliary graph, call it $\delta_F(S^*)$; note that S^* is nonempty and $S^* \subsetneq V$. Let μ denote the capacity of $\delta_F(S^*)$; thus, $\mu = (p+q)\,|\delta_F(S^*)\cap \mathcal{S}| + p\,|\delta_F(S^*)\cap \mathcal{U}|$. If $\mu < p(p+q)$, then F is infeasible. Otherwise, we compute the set $\widehat{\mathcal{C}}$ of all cuts of the auxiliary graph that have capacity between μ and 2μ, by applying the polynomial-time algorithm of Nagamochi, Nishimura and Ibaraki [16] that enumerates over all 2-approximate minimum-cuts of a capacitated graph. We exhaustively check whether or not each of the cuts in $\widehat{\mathcal{C}}$ has either $(p+q)$ edges or has p safe edges. Clearly, F is infeasible if $\widehat{\mathcal{C}}$ contains a cut that violates this condition. Otherwise, F is feasible because any cut $\delta_F(S)$, $\emptyset \neq S \subsetneq V$, of the auxiliary graph that is not in $\widehat{\mathcal{C}}$ has capacity $\geq 2\,\mu \geq 2\,p(p+q)$, and so either $(p+q)\,|\delta_F(S)\cap \mathcal{S}| \geq p(p+q)$, that is, $\delta_F(S)$ has $\geq p$ safe edges, or $p\,|\delta_F(S)\cap \mathcal{U}| \geq p(p+q)$, that is, $\delta_F(S)$ has $\geq p+q$ (unsafe) edges. □

For the rest of this section, we assume that the given instance of (p,q)-FGC is feasible. Let F^* denote an optimal solution for the (p,q)-FGC instance, and let $\mathrm{OPT} = c(F^*)$ denote the optimal value.

Given an edge-set F (i.e., a candidate solution), we call a cut $\delta(S)$, $\emptyset \neq S \subsetneq V$, *deficient* if $|F\cap\delta(S)\cap\mathcal{S}| < p$ and $|F\cap\delta(S)| < p+q$; thus, a deficient cut is one that certifies the infeasibility of F.

First, we give an overview of our $O(q\log n)$-approximation algorithm for (p,q)-FGC. Our algorithm runs in two stages. In the first stage, we construct an instance of the Cap-k-ECSS problem (that partially models the given (p,q)-FGC instance), and then we apply our approximation algorithm for the Cap-k-ECSS problem (see Theorem 1.3) to compute an edge-set F of cost $O(q\cdot\mathrm{OPT})$ that is "nearly feasible" for the (p,q)-FGC instance. In more detail, the set of cuts that are deficient w.r.t. F has size polynomial in n, and the set is computable in time polynomial in n. The second stage of our algorithm applies several iterations. In each iteration $\ell = 1,2,\ldots$, we find all the deficient cuts of our current subgraph $H=(V,F)$ and then we apply the greedy algorithm for the (well-known) hitting-set problem to find an edge-set F_ℓ that covers all the deficient cuts (i.e., each of the deficient cuts contains at least one edge of F_ℓ). Then, we update F to $F\,\dot\cup\,F_\ell$, i.e., we augment F by the edge-set computed by the greedy algorithm, and then we re-compute the set of deficient cuts w.r.t. the updated subgraph $H=(V,F)$. We stop iterating when there are no deficient cuts w.r.t. $H=(V,F)$; thus, at the termination of the second stage, F is feasible for the (p,q)-FGC instance. The greedy algorithm (for the hitting-set instances that arise) achieves an approximation factor of $O(\log n)$.

We discuss the hitting-set problem and state the approximation factor of the greedy algorithm for this problem.

Definition 5.1 (Hitting-Set Problem)**.** *Given a ground set $\mathcal{E} = \{e_1, \ldots, e_{\hat{n}}\}$ of elements, a family $\mathcal{R} \subseteq 2^{\mathcal{E}}$ of subsets of $\mathcal{E}$, and nonnegative costs $c : \mathcal{E} \to \mathbb{R}_{\geq 0}$ on the elements of $\mathcal{E}$, find a minimum-cost subset $F \subseteq \mathcal{E}$ such that for every $R \in \mathcal{R}$, we have $F \cap R \neq \emptyset$.*

It is well-known that there is an $O(\log |\mathcal{R}|)$-approximation algorithm for the hitting-set problem based on the greedy strategy, see [21].

Proposition 5.2 ([21])**.** *There is an $O(\log |\mathcal{R}|)$-approximation algorithm for the hitting-set problem that runs in time polynomial in $|\mathcal{E}|$ and $|\mathcal{R}|$.*

Constructing an instance of Cap-k-ECSS**:** Given an instance of (p, q)-FGC, we construct an instance of the Cap-k-ECSS problem on the underlying graph $G = (V, E)$ with the same edge costs $c : E \to \mathbb{R}_{\geq 0}$.

Recall that our plan is to apply the $\min(k, 2u_{\max})$-approximation algorithm for Cap-k-ECSS (Theorem 1.3) to compute a cheap edge-set F that is "nearly feasible" for the (p, q)-FGC instance. We consider two cases, one for $p > q$, and the other for $p \leq q$, and we use different values of the edge capacities for the two cases. In the first case, we use unit edge capacities, and in the second case, we use the edge capacities given in the proof of Proposition 5.1. Formally:

Case 1 $p > q$**:** $k := p$, and all edges have unit capacity, i.e., $u_e := 1$ for all $e \in E$.

Case 2 $p \leq q$**:** $k := p(p + q)$, each safe edge has capacity $(p + q)$, and each unsafe edge has capacity p, that is, $u_e = p + q$ if $e \in \mathcal{S}$, and $u_e = p$ if $e \in \mathcal{U}$.

Let $u_{\max} = \max_{e \in E} u_e$ be the maximum edge capacity.

First, we argue that the instance of Cap-k-ECSS is feasible. Recall that F^* denotes an optimal solution for the given (feasible) (p, q)-FGC instance. By Proposition 5.1, for any nonempty set $S \subsetneq V$, we either have $|\delta(S) \cap F^* \cap \mathcal{S}| \geq p$ or $|\delta(S) \cap F^*| \geq p + q$. The feasibility of the Cap-k-ECSS instance follows from showing that for all nonempty sets $S \subsetneq V$, we have $u(\delta(S)) \geq k$. To this end, fix such an S and recall the choice of edge capacities that depends on the relative values of p and q. If $p > q$, then

$$u(\delta(S)) = |\delta(S)| \geq |\delta(S) \cap F^*| \geq p = k. \tag{5:1}$$

On the other hand, if $p \leq q$, then we have:

(5:2)

$$u(\delta(S)) \geq u(\delta(S) \cap F^*) \geq \max((p+q) \cdot |\delta(S) \cap F^* \cap \mathcal{S}|, p \cdot |\delta(S) \cap F^*|) \geq p(p+q) = k.$$

Let $F \subseteq E$ be a feasible solution to the Cap-k-ECSS instance; thus, every nonempty $S \subsetneq V$ has $u(\delta(S) \cap F) \geq k$. Let

$$\mathcal{C} := \{S \subsetneq V : S \neq \emptyset, u(\delta(S) \cap F) \leq 2k, |\delta(S) \cap F| < p + q, |\delta(S) \cap F \cap \mathcal{S}| < p\}.$$

Informally speaking, in the capacitated graph $H = (V, F, u)$ that corresponds to the edge-set F, $\mathcal{C}$ is the collection of all vertex-sets that correspond to 2-approximate minimum-cuts that violate the feasibility requirement stated in Proposition 5.1.

Description of our two-stage algorithm for (p, q)**-FGC:** In the first stage of the algorithm, we use Theorem 1.3 to obtain a feasible solution F for the above Cap-k-ECSS instance such that the cost $c(F)$ is $\leq \min(k, 2u_{\max}) \cdot \mathrm{OPT}$. The second stage of the algorithm consists of several iterations that augment edges to F until all the deficient cuts w.r.t. F have been fixed. The ℓ^{th} iteration (for some $\ell = 1, 2, \dots$) consists of solving an instance of the hitting-set problem: we want to hit all sets in the collection $\{\delta(S) \cap (E \setminus F) \ : \ S \in \mathcal{C}\}$ by using edge-elements $e \in E \setminus F$, where the cost of e is c_e (in the hitting-set instance). Let F'_ℓ denote a hitting-set computed by the greedy algorithm for the above hitting-set instance. We update F to $F \dot{\cup} F'_\ell$, and recompute $\mathcal{C}$ using the new F. As long as $\mathcal{C}$ is not empty, we repeat the above iteration. When $\mathcal{C}$ becomes empty, we return the current F as a feasible solution of the given (p, q)-FGC instance. Assuming that the number of iterations in the second stage is $O(q)$, the cost of F is $O(q \cdot \mathrm{OPT} + O(q \log n) \cdot \mathrm{OPT}) = O(q \log n) \cdot \mathrm{OPT}$.

Observe that each of the hitting-set instances constructed by the algorithm is feasible, because for any deficient cut $\delta(S)$ w.r.t. the current solution F, we have $(F^* \setminus F) \cap \delta(S) \neq \emptyset$, that is, $F^* \setminus F$ is a feasible hitting-set.

The next result shows that the algorithm finds a feasible solution.

Lemma 5.3. *The edge-set F returned by the algorithm is feasible for the given* (p, q)-FGC *instance.*

Proof. By the design of the second stage of the algorithm, the solution F is repeatedly augmented as long as $\mathcal{C}$ is nonempty, so it suffices to argue that every deficient cut (w.r.t. the current F) belongs to $\mathcal{C}$. Let $\delta(S)$, $\emptyset \neq S \subsetneq V$, be an arbitrary deficient cut w.r.t. F. By definition, $|\delta(S) \cap F \cap \mathcal{S}| < p$ and $|\delta(S) \cap F| < p + q$. To show that S belongs to $\mathcal{C}$, we need to show that $u(\delta(S) \cap F) \leq 2k$ holds. If $p > q$, then

$$u(\delta(S) \cap F) = |\delta(S) \cap F| < p + q < 2p = 2k. \tag{5:3}$$

On the other hand, if $p \leq q$, then

$$u(\delta(S) \cap F) = p \cdot |\delta(S) \cap F| + q \cdot |\delta(S) \cap F \cap \mathcal{S}| < p(p+q) + pq < 2p(p+q) = 2k. \tag{5:4}$$

Thus, in either case, deficient cuts are 2-approximate minimum-cuts in the capacitated graph $H = (V, F, u)$, and they belong to $\mathcal{C}$. □

We complete the proof of Theorem 1.5 by arguing that the above algorithm finds a feasible solution of the (p, q)-FGC instance of cost $O(q \log n) \cdot \mathrm{OPT}$ in polynomial time.

Lemma 5.4. *The above two-stage algorithm runs in time polynomial in n and returns a feasible solution $F \subseteq E$ for* (p, q)-FGC *such that $c(F) \leq O(q \log n) \cdot \mathrm{OPT}$.*

Proof. We argue that the output of the algorithm has cost at most $O(q \log n) \cdot \mathrm{OPT}$. It is easy to see that the cost of the first-stage solution is $O(q) \cdot \mathrm{OPT}$ because of

the approximation factor from Theorem 1.3; as mentioned before, F^* is feasible for the Cap-k-ECSS instance, and $\min(k, u_{\max}) \leq 2q$, for all relevant values of p and q. We bound the cost incurred in the second stage by arguing that: (i) each iteration leads to an additional cost of at most $O(\log n) \cdot \mathrm{OPT}$; and (ii) there are at most q iterations.

In the capacitated graph $H(V, F, u)$, all deficient cuts are 2-approximate minimum-cuts, i.e., the capacity of each deficient cut is at most two times the capacity of a minimum-cut. Karger's bound [13] on the number of 2-approximate minimum-cuts implies that $|\mathcal{C}| = O(n^4)$. By using the algorithm of Nagamochi et al. [16] we can explicitly compute the collection $\mathcal{C}$ in (deterministic) polynomial time. Since $F^* \setminus F$ is a feasible solution to each of the hitting-set instances constructed in the second stage, Proposition 5.2 implies that the cost of the augmenting edge-set found in each iteration is $O(\log |\mathcal{C}|) \cdot c(F^* \setminus F) = O(\log n) \cdot \mathrm{OPT}$, thereby showing (i).

Next, we bound the number of iterations via a case analysis. First, suppose that $p > q$. Then each iteration increases the capacity of a deficient cut by at least one. By (5:1), every deficient cut has capacity at least p at the end of the first stage, and due to (5:3), a cut is no longer deficient once its capacity is at least $p + q$. Next, suppose that $p \leq q$. We use a similar argument for this case. Now, each iteration increases the capacity of a deficient cut by at least p. By (5:2), every deficient cut has capacity at least $p(p + q)$ at the end of the first stage, and due to (5:4), a cut is longer deficient once its capacity is at least $p(p + q) + pq$. Overall, we have $c(F) \leq (2q + q \cdot O(\log n)) \cdot \mathrm{OPT} = O(q \log n) \cdot \mathrm{OPT}$, as desired.

Lastly, we argue that the entire algorithm runs in polynomial time. The first stage runs in (deterministic) polynomial time, by Theorem 1.3. The second stage also runs in (deterministic) polynomial time because the number of iterations is at most q ($\leq n$), and in each iteration we solve a polynomial-sized hitting-set instance. This completes the proof of the lemma. □

6. Unweighted problems: $(1, 1)$-FGC and $(k, 1)$-FGC

Consider the unweighted version of FGC where each edge has unit cost, i.e., $c_e = 1$ for all $e \in E$. We present a $\frac{16}{11}$-approximation algorithm (see Theorem 1.6). To the best of our knowledge, this is the first result that improves on the approximation factor of $\frac{3}{2}$ for unweighted FGC. In fact, we give two algorithms for obtaining two candidate solutions to an instance of unweighted FGC; the simpler of these algorithms is discussed by Adjiashvili et al. [1, 2]. Assuming that we have access to an α-approximation algorithm for the minimum-size (i.e., unweighted) 2-ECSS problem, we argue that the cheaper of the two candidate solutions is a $\frac{4\alpha}{2\alpha+1}$-approximate solution to the instance of unweighted FGC. Adjiashvili et al. [2] gave an $\left(\frac{\alpha}{2} + 1\right)$-approximation algorithm for unweighted FGC, assuming the existence of an α-approximation algorithm for the minimum-size (i.e., unweighted) 2-ECSS problem; this implies a $\frac{5}{3}$-approximation algorithm for unweighted FGC by using the results of [19, 11]. The algorithm in [2] starts with a maximal forest of safe edges in the graph. At the end of this section, we give an example showing

that the (asymptotic) approximation factor achievable by such an algorithm is at least $\frac{3}{2}$. Our main result in this section is the following.

Theorem 6.1. *Suppose that there is an α-approximation algorithm for the minimum-size (i.e., unweighted)* 2-ECSS *problem. Then, there is a $\frac{4\alpha}{2\alpha+1}$-approximation algorithm for unweighted* FGC*.*

Theorem 1.6 follows from the above theorem by using the $\frac{4}{3}$-approximation algorithm of [19, 11] for the minimum-size 2-ECSS problem. Before delving into the proof of Theorem 6.1, we introduce some basic results on W-joins, which will be useful in our algorithm and its analysis. Let $G' = (V', E')$ be an undirected graph with no self-loops and let $c' : E' \to \mathbb{R}_{\geq 0}$ be nonnegative costs on the edges.

Definition 6.1 (W-join)**.** *Let $W \subseteq V'$ be a subset of vertices with $|W|$ even. A subset $J \subseteq E'$ of edges is called a W-join if W is equal to the set of vertices of odd degree in the subgraph (V', J).*

The following classical result on finding a minimum-cost W-join is due to Edmonds.

Proposition 6.2 ([18], Theorem 29.1)**.** *Given (G', c'), we can either obtain a minimum-cost W-join, or conclude that there is no W-join, in strongly polynomial time.*

The W-join polytope is the convex hull of the incidence vectors of W-joins. Edmonds & Johnson showed that the dominant of the W-join polytope has a simple linear description.

Proposition 6.3 ([18], Corollary 29.2b)**.** *The dominant of the W-join polytope is given by $\{x \in \mathbb{R}^{E'}_{\geq 0} : x(\delta_{G'}(S)) \geq 1 \,\forall\, S \subsetneq V' \text{ s.t. } |S \cap W| \text{ odd}\}$.*

Consider an instance of unweighted FGC consisting of a graph $G = (V, E)$ with a specified partition of E into a set of safe edges and a set of unsafe edges, $E = \mathcal{S} \dot\cup \mathcal{U}$. We will assume that G is connected and has no unsafe bridges (i.e., cut-edges), since otherwise the instance is infeasible. Let F^* denote an optimal solution.

Join-based algorithm for unweighted FGC**:** Let T be a spanning tree in G that maximizes the number of safe edges. Clearly, for each safe edge $e = uv$ in $E(G) - T$, the (unique) u, v-path in T consists of safe edges; hence, the graph obtained from G by contracting all the safe edges of T has no safe edges. If $|T \cap \mathcal{S}| = |V| - 1$, then T is an optimal FGC solution for the given instance, and we are done. Otherwise, let $T' := T \cap \mathcal{U}$ be the (nonempty) set of unsafe edges in T. Let $G' = (V', E')$ denote the graph obtained from G by contracting all the (safe) edges in $T \setminus T'$. (We remove all self-loops from G', but retain parallel edges that arise due to edge contractions.) Note that all edges in E' are unsafe (by the discussion above), and T' is a spanning tree of G'. Let W' denote the (nonempty) set of odd degree vertices in the subgraph (V', T'). Using Proposition 6.2, in polynomial time, we compute a minimum-cardinality W'-join in G', and denote it by $J' \subseteq E'$. By our choice, the subgraph $(V', T' \dot\cup J')$ is connected and Eulerian, so it is 2-edge connected in G'.

Consider the multiset $F_1 = T \dot\cup J'$ consisting of edges in E; if an edge e appears in both T' and J', then we include two copies of e in F_1.
If F_1 contains at most one copy of each edge in E, then F_1 is FGC-feasible. Otherwise, we modify F_1 to get rid of all duplicates without increasing $|F_1|$. Consider an unsafe edge $e \in E'$ that appears twice in F_1, i.e., e belongs to both T' and J'. We remove a copy of e from F_1. If this does not violate FGC-feasibility, then we take no further action. Otherwise, the second copy of e in F_1 is an unsafe bridge in (V, F_1) that induces a cut $\delta(S)$, $\emptyset \neq S \subsetneq V$, in G. By our assumption that G has no unsafe bridges, there is another edge $e' \in E$ that is in $\delta(S)$ but not in F_1. We include e' in F_1. This finishes the description of our first algorithm.

At the end of the de-duplication step, F_1 is FGC-feasible and it contains at most one copy of any edge $e \in E$. It is also clear that $|F_1| \leq |T| + |J'|$. We claim that $|J'| \leq \frac{1}{2}|F^* \cap \mathcal{U}|$. In fact, this follows from a result of Frank [5] (also, see [19, Proposition 6]); Frank's result states that the minimum size of a W-join (for any set of vertices W with $|W|$ even) is at most half the minimum size of a 2-ECSS; observe that J' is a W'-join of the graph G' and $F^* \cap \mathcal{U}$ is a 2-ECSS of G', hence, the inequality follows from Frank's result. We give another proof of the inequality, for the sake of completeness.

Claim 6.4. *We have* $|J'| \leq \frac{1}{2}|F^* \cap \mathcal{U}|$. *Hence,* $|F_1| \leq |F^* \cap \mathcal{S}| + \frac{3}{2}|F^* \cap \mathcal{U}|$.

Proof. We prove the claim by constructing a fractional W'-join of small size. Recall that we chose T such that $|T \cap \mathcal{S}|$ is maximum, and we obtained G' by contracting connected components in $(V, T \setminus T')$. G' consists of only unsafe edges, and moreover, G' is 2-edge connected because G has no unsafe bridges (by our assumption). Let $B := F^* \cap E'$ denote the set of unsafe edges in the optimal solution F^* that also belong to G'. Consider the vector $z := \frac{1}{2}\chi^B$ where $\chi^B \in [0,1]^{E'}$ is the incidence vector of B in G'. Let $\delta(S')$, $\emptyset \neq S' \subsetneq V(G')$, be an arbitrary cut in G' and let $\delta(S)$ be the unique cut in G that gives rise to $\delta(S')$ when we contract (safe) edges in $T \setminus T'$. Since F^* is FGC-feasible and there are no safe edges in $\delta_G(S)$, we must have $|B \cap \delta_{G'}(S')| \geq 2$. Consequently, $z(\delta_{G'}(S')) = \frac{1}{2}|B \cap \delta_{G'}(S')| \geq 1$. By Proposition 6.3, z lies in the dominant of the W'-join polytope, i.e., z dominates a fractional W'-join. Since J' is a min-cardinality W'-join, $|J'| \leq \mathbf{1}^T z \leq \frac{1}{2}|F^* \cap \mathcal{U}|$. We bound the size of F_1 by using the trivial bound $|T| \leq |F^*|$:

$$|F_1| \leq |F^*| + |J'| \leq |F^* \cap \mathcal{S}| + \frac{3}{2}|F^* \cap \mathcal{U}|. \qquad \square$$

Our second algorithm uses the α-approximation for the minimum-size 2-ECSS problem as a subroutine. Informally speaking, the solution returned by this algorithm has the property that its size complements that of F_1.

2-ECSS-based algorithm for unweighted FGC: Consider the graph G'' obtained from G by duplicating every safe edge in E. Similarly, let F'' be the multiedge-set obtained from F^* by duplicating every safe edge in F^*. Clearly, (V, F'') is a 2-edge connected subgraph of G'' consisting of $2|F^* \cap \mathcal{S}| + |F^* \cap \mathcal{U}|$ edges. Let F_2 be the output of running the α-approximation algorithm for the minimum-size 2-ECSS problem on G''. Since F_2 is 2-edge connected and only safe edges can appear

more than once in F_2 (because G'' only has duplicates of safe edges), we can drop the extra copy of all safe edges while maintaining FGC-feasibility in G. This finishes the description of our second algorithm.

The following claim is immediate.

Claim 6.5. *We have* $|F_2| \leq 2\alpha|F^* \cap \mathcal{S}| + \alpha|F^* \cap \mathcal{U}|$.

We end this section with the proof of our main result on unweighted FGC.

Proof of Theorem 6.1. Given an instance of unweighted FGC, we compute two candidate solutions F_1 and F_2 as given by the two algorithms described above. The solution F_1 can be computed using algorithms for the MST problem and the minimum-weight W'-join problem, followed by basic graph operations. The solution F_2 can be computed using the given α-approximation algorithm for the minimum-size 2-ECSS problem. We show that the smaller of F_1 and F_2 is a $\frac{4\alpha}{2\alpha+1}$-approximate solution for the instance of unweighted FGC. By Claims 6.4 and 6.5:

$$\min(|F_1|, |F_2|) \leq \frac{2\alpha}{2\alpha+1}|F_1| + \frac{1}{2\alpha+1}|F_2| = \frac{4\alpha}{2\alpha+1}|F^*| \qquad \square$$

As mentioned earlier, we have an example (see Figure 1 below) such that any algorithm for unweighted FGC that starts with a maximal forest on safe edges achieves an (asymptotic) approximation factor of $\frac{3}{2}$ or more.

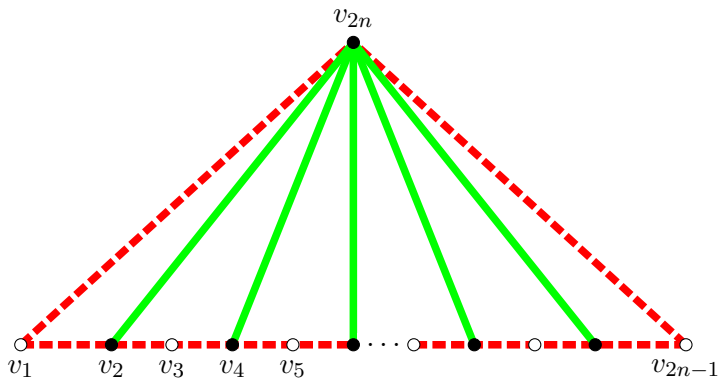

FIGURE 1. In this instance we have a graph on $2n$ vertices. The set of unsafe edges, shown using dashed red lines, forms a Hamiltonian cycle. For each $i = 1, \ldots, n-1$, there is a safe edge, shown using a solid green line, between v_{2i} and v_{2n}. The solution consisting of all unsafe edges is feasible, and any feasible solution must contain all unsafe edges, so the value of an optimal integral solution is $2n$. Any feasible solution that contains a maximal forest on safe edges has size at least $3n - 1$.

Improved approximation factor for unweighted $(k,1)$**-FGC:** Finally, we focus on unweighted $(k,1)$-FGC where each edge has unit cost. We can improve on the approximation factor of four of Theorem 1.4, by using the same method (see Section 4), except that in the first stage we apply the best approximation algorithm known for the minimum-size (unweighted) k-ECSS problem. Let α_k denote the best approximation factor known for the latter problem. Note that $\alpha_2 = 4/3$ (see [19, 11]), $\alpha_3 = 1.5$ (see Gabow [6]), and, in general, $\alpha_k < 1 + \frac{1.91}{k}$ (see Gabow & Gallagher [7]).

Proposition 6.6. *There is a* $(2 + \alpha_k)$*-approximation algorithm for unweighted* $(k,1)$-FGC. *Thus, the approximation factor is* $\frac{10}{3}$ *for* $k = 2$, $\frac{7}{2}$ *for* $k = 3$, *and it is less than* $(3 + \frac{1.91}{k})$ *for all integers* $k \geq 4$.

7. Appendix: Examples of $(2,2)$-FGC and Cap-4-ECSS

In this section, we illustrate the notions of (p,q)-FGC and Cap-k-ECSS through an explicit example. Consider the graph G given in Figure 2.

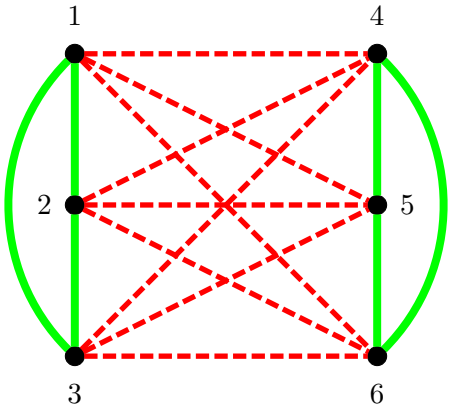

FIGURE 2. The graph $G = (\{1,2,3,4,5,6\}, E)$ shown above is a union of three subgraphs L, R, B. The subgraphs $L = (\{1,2,3\}, \{12, 23, 31\})$ and $R = (\{4,5,6\}, \{45, 56, 64\})$ are complete graphs whose edges are all safe (depicted using solid green edges) and have both cost and capacity equal to two. The subgraph B is a complete bipartite graph on the bipartition $\{1,2,3\} \sqcup \{4,5,6\}$ whose edges are all unsafe (depicted using dashed red edges) and have both cost and capacity equal to one.

We first consider a (p,q)-FGC instance on the graph G with $p = q = 2$. It is easy to see that the spanning subgraph H_1 comprising of all (safe) edges from $E(L) \cup E(R)$ and any four unsafe edges from B is feasible for the $(2,2)$-FGC problem (see Proposition 5.1). Note that the cost of H_1 is $6 \cdot 2 + 4 \cdot 1 = 16$. In Claim 7.1 (see below), we show that H_1 is an optimal solution.

We next consider a Cap-k-ECSS instance on the graph G with $k = 4$. Consider the spanning subgraph H_2 comprising of capacity-2 edges $\{12, 23, 45, 56\}$ and unit-capacity edges $\{14, 42, 26, 63, 35, 51\}$. We claim that H_2 is feasible for the Cap-4-ECSS instance. To see this, observe that the set of unit-capacity edges in H_2 forms a Hamiltonian cycle C, so the capacity of every cut $\delta_C(S)$, $\emptyset \subsetneq S \subsetneq V$, is a positive even number. Furthermore, $u(\delta_C(S)) = 2$ if and only if S consists of consecutive vertices from the cycle $(1,4,2,6,3,5,1)$. Since $\{12, 23\}$ and $\{45, 56\}$ form spanning trees in L and R, respectively, $\delta_{H_2}(S)$ contains at least one capacity-2 edge. Thus, $u(\delta_{H_2}(S)) \geq 4$ for all $\emptyset \subsetneq S \subsetneq V$. Clearly, the cost of H_2 is $4 \cdot 2 + 6 \cdot 1 = 14$.

By our choice of edge-capacities and k, it is easy to see that any feasible subgraph for the $(2,2)$-FGC instance is also feasible for the Cap-4-ECSS instance. The following claim shows that the optimal value of the former instance is strictly larger than the optimal value of the latter instance.

Claim 7.1. *Any feasible solution to the above* $(2,2)$-FGC *instance has cost* ≥ 16.

Proof. Fix some feasible solution $F \subseteq E$. First, observe that $|F \cap E(B)| \geq 4$, i.e., F contains at least four unsafe edges; this holds because the cut $\delta_F(\{1,2,3\})$ has $\geq p+q=4$ edges, see Proposition 5.1. Clearly, if F contains all six safe edges, then $c(F) \geq 16$ follows immediately. Now, we assume that F has ≤ 5 safe edges.

Next, we argue that both inequalities $|F \cap E(L)| \geq 2$ and $|F \cap E(R)| \geq 2$ hold. By symmetry, it suffices to prove one of these inequalities. Suppose, for the sake of contradiction, that $|F \cap E(L)| \leq 1$ holds. Again, by symmetry, we may assume that either $F \cap E(L) = \emptyset$ or $F \cap E(L) = \{12\}$. Note that F has ≤ 3 edges incident to vertex 3 and all edges of $\delta_F(\{3\})$ are unsafe, so deleting two of these edges results in a graph that has ≤ 1 edge incident to vertex 3; clearly, such a graph is not 2-edge connected. Hence, F has ≥ 2 safe edges from each of $E(L)$ and $E(R)$.

By symmetry, suppose that $|F \cap E(L)| = 2 \leq |F \cap E(R)|$. Then, there are two vertices $s, t \in \{1,2,3\}$ that are each incident to exactly one safe edge. Each of these vertices must be incident to 3 unsafe edges, otherwise, by deleting all of the unsafe edges incident to either s or t, we obtain a graph that has a vertex of degree one, i.e., deleting two unsafe edges from F results in a graph that is not 2-edge connected. It follows that either F has 5 safe edges and ≥ 6 unsafe edges, or else F has 4 safe edges and ≥ 8 unsafe edges. In either case, we have $c(F) \geq 16$. □

Acknowledgements. We thank the anonymous reviewers and PC members of FSTTCS 2021 for their comments.

References

[1] D. Adjiashvili, F. Hommelsheim, and M. Mühlenthaler. Flexible Graph Connectivity. In *Proceedings of the 21st Integer Programming and Combinatorial Optimization Conference*, volume 12125 of *Lecture Notes in Computer Science*, pages 13–26, 2020.

[2] D. Adjiashvili, F. Hommelsheim, and M. Mühlenthaler. Flexible Graph Connectivity. *Mathematical Programming*, pages 1–33, 2021.

[3] D. Chakrabarty, C. Chekuri, S. Khanna, and N. Korula. Approximability of Capacitated Network Design. *Algorithmica*, 72(2):493–514, 2015.

[4] L. Fleischer. Building Chain and Cactus Representations of All Minimum Cuts from Hao-Orlin in the Same Asymptotic Run Time. *Journal of Algorithms*, 33(1):51–72, 1999.

[5] A. Frank. Conservative weightings and ear-decompositions of graphs. *Combinatorica*, 13(1):65–81, 1993.

[6] H. N. Gabow. An Ear Decomposition Approach to Approximating the Smallest 3-Edge Connected Spanning Subgraph of a Multigraph. *SIAM Journal on Discrete Mathematics*, 18(1):41–70, 2004.

[7] H. N. Gabow and S. Gallagher. Iterated Rounding Algorithms for the Smallest k-Edge Connected Spanning Subgraph. *SIAM Journal on Computing*, 41(1):61–103, 2012.

[8] H. N. Gabow, M. X. Goemans, É. Tardos, and D. P. Williamson. Approximating the Smallest k-Edge Connected Spanning Subgraph by LP-Rounding. *Networks*, 53(4):345–357, 2009.

[9] M. X. Goemans, A. V. Goldberg, S. A. Plotkin, D. B. Shmoys, É. Tardos, and D. P. Williamson. Improved Approximation Algorithms for Network Design Problems. In *Proceedings of the 5th Symposium on Discrete Algorithms*, pages 223–232, 1994.

[10] M. X. Goemans and D. P. Williamson. *The Primal-Dual Method for Approximation Algorithms and its Application to Network Design Problems*, chapter 4, pages 144–191. PWS Publishing Company, Boston, MA, 1997.

[11] C. Hunkenschröder, S. Vempala, and A. Vetta. A 4/3-Approximation Algorithm for the Minimum 2-Edge Connected Subgraph Problem. *ACM Transactions on Algorithms*, 15(4):1–28, 2019.

[12] K. Jain. A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. *Combinatorica*, 21(1):39–60, 2001.

[13] D. R. Karger. Global Min-Cuts in $\mathcal{RNC}$, and Other Ramifications of a Simple Min-Cut Algorithm. In *Proceedings of the 4th Symposium on Discrete Algorithms*, pages 21–30, 1993.

[14] S. Khuller and U. Vishkin. Biconnectivity Approximations and Graph Carvings. *Journal of the ACM*, 41(2):214–235, 1994.

[15] T. L. Magnanti and R. T. Wong. Network Design and Transportation Planning: Models and Algorithms. *Transportation Science*, 18(1):1–55, 1984.

[16] H. Nagamochi, K. Nishimura, and T. Ibaraki. Computing All Small Cuts in an Undirected Network. *SIAM Journal on Discrete Mathematics*, 10(3):469–481, 1997.

[17] P. Rozenshtein, A. Gionis, B. A. Prakash, and J. Vreeken. Reconstructing an Epidemic Over Time. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, pages 1835–1844, 2016.

[18] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag Berlin Heidelberg, 2003.

[19] A. Sebö and J. Vygen. Shorter Tours by Nicer Ears: 7/5-Approximation for the Graph-TSP, 3/2 for the Path Version, and 4/3 for Two-Edge-Connected Subgraphs. *Combinatorica*, 34(5):597–629, 2014.

[20] L. V. Snyder, M. P. Scaparra, M. S. Daskin, and R. L. Church. *Planning for Disruptions in Supply Chain Networks*, pages 234–257. Institute for Operations Research and the Management Sciences, 2014.

[21] V. V. Vazirani. *Approximation Algorithms*. Springer Berlin Heidelberg, 2003.

[22] 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.

School of Electrical Engineering & Computer Science, University of Ottawa, Canada
E-mail address: sboyd@uottawa.ca
URL: https://www.site.uottawa.ca/~sylvia

Department of Combinatorics and Optimization, University of Waterloo, Canada
E-mail address: jcheriyan@uwaterloo.ca
URL: https://www.math.uwaterloo.ca/~jcheriyan

Modeling and Optimization, Amazon Inc., Bellevue, WA, USA
E-mail address: arash.haddadan@gmail.com

Department of Mathematics, London School of Economics and Political Science, UK
E-mail address: S.Ibrahimpur@lse.ac.uk
URL: https://www.math.uwaterloo.ca/~s26ibrah https://orcid.org/0000-0002-1575-9648