Approximation Algorithms for Flexible Graph Connectivity

Sylvia Boyd
School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Joseph Cheriyan
Department of Combinatorics and Optimization, University of Waterloo, Canada

Arash Haddadan
Warner Music Group, New York, NY, USA

Sharat Ibrahimpur
Department of Combinatorics and Optimization, University of Waterloo, Canada

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). In an instance of the Flexible Graph Connectivity (FGC) problem, 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 $\{c_e\}_{e \in E}$ on the edges. 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 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 FGC via a reduction to the minimum-cost r-out 2-arborescence problem. This improves upon the 2.527-approximation algorithm of Adjiashvili et al.

For integers $p \geq 1$ and $q \geq 0$, the (p, q)-FGC problem is a generalization of FGC where we seek a minimum-cost subgraph $H = (V, F)$ that remains p-edge connected against the failure of any set of at most q unsafe edges; that is, for any set $F' \subseteq \mathcal{U}$ with $|F'| \leq q$, $H - F' = (V, F \setminus F')$ should be p-edge connected. Note that FGC corresponds to the $(1, 1)$-FGC problem. We give approximation algorithms for two important special cases of (p, q)-FGC: (a) Our 2-approximation algorithm for FGC extends to a $(k + 1)$-approximation algorithm for the $(1, k)$-FGC problem. (b) We present a 4-approximation algorithm for the $(k, 1)$-FGC problem.

For the unweighted FGC problem, where each edge has unit cost, we give a 16/11-approximation algorithm. This improves on the result of Adjiashvili et al. for this problem.

The (p, q)-FGC model with $p = 1$ or $q \leq 1$ can be cast as the *Capacitated k-Connected Subgraph* problem which is a special case of the well-known Capacitated Network Design problem. We denote the former problem by Cap-k-ECSS. An instance of this problem consists of an undirected graph $G = (V, E)$, nonnegative integer edge-capacities $\{u_e\}_{e \in E}$, nonnegative edge-costs $\{c_e\}_{e \in E}$, and a positive integer k. The goal is to find a minimum-cost edge-set $F \subseteq E$ such that every (non-trivial) cut of the capacitated subgraph $H(V, F, u)$ has capacity at least k. We give a $\min(k, 2 \max_{e \in E} u_e)$-approximation algorithm for this problem.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

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

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.9

Funding *Joseph Cheriyan*: Supported in part by NSERC, RGPIN-2019-04197.
Arash Haddadan: This work was mostly done when this author was a postdoctoral researcher at the Biocomplexity Institute and Initiative at the University of Virginia, Charlottesville, and supported by NSF Expeditions in Computing Grant with award number CCF-1918656.
Sharat Ibrahimpur: Supported in part by NSERC grant 327620-09.

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

41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021).
Editors: Mikołaj Bojańczyk and Chandra Chekuri; Article No. 9; pp. 9:1–9:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 from practical considerations such as the design of fault-tolerant supply chains, congestion control for urban road traffic, and the modeling of epidemics (see [11, 12, 15]).

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 an instance of FGC, we have an undirected connected graph $G=(V,E)$ on n vertices, a partition of E into safe edges $\mathcal{S}$ and unsafe edges $\mathcal{U}$, and nonnegative costs $\{c_e\}_{e\in E}$ on the edges. The graph G may have multiedges, but no self-loops. 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) solution F minimizing $c(F)=\sum_{e\in F}c_e$. 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 2ECSS) corresponds to an instance of FGC where all edges are unsafe, and the MST problem corresponds to an instance of FGC where all edges are safe. Second, FGC captures a non-uniform model of survivable network design problems where a subset of edges never fail, i.e., they are always safe.

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 $q\geq 0$. 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 nontrivial cut $(S,V\setminus S)$ of H either contains p safe edges or contains $p+q$ (safe or unsafe) edges. Note that the FGC problem is the same as the $(1,1)$-FGC problem. 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. It can be seen as a way of interpolating between p-edge connectivity (when all edges are safe) and $(p+q)$-edge connectivity (when all edges are unsafe). We remark that for all problems considered in this work, we are only allowed to use at most one copy of an edge; multiedges may arise in F due to multiedges in G.

One of our goals is to give approximation algorithms for important special cases of (p,q)-FGC. Since FGC generalizes the 2ECSS problem, it is already APX-hard (see [5]), so a polynomial-time approximation scheme is ruled out unless P=NP. In the following we sketch a simple randomized $O(q\log n)$-approximation algorithm for (p,q)-FGC under some assumptions. For simplicity, assume that p and q are such that $q/p\leq\alpha$ for an absolute constant $\alpha\geq 0$. Let F^* denote an optimal solution to the given (p,q)-FGC instance. To start with, let $H=(V,F)$ denote a 2-approximate p-edge connected spanning subgraph (abbreviated p-ECSS) of the graph G with edge-costs $\{c_e\}_{e\in E}$, where we make no distinction between safe and unsafe edges; such an H can be found in polynomial-time by using (say) Jain's iterative rounding algorithm [8]. Note that $c(F)\leq 2c(F^*)$, since (V,F^*) is a p-edge connected spanning subgraph of G.

We say that a nonempty set $R \subsetneq V$ is "deficient" if $|\delta(R) \cap F| < p+q$ and $|\delta(R) \cap F \cap \mathcal{S}| < p$; thus, $R \subsetneq V$ is deficient if the cut $\delta(R)$ has less than $(p+q)$ F-edges and has less than p safe F-edges. Let $\mathcal{C}$ denote the family of all deficient sets. Note that deficient sets are the only obstructions to the (p, q)-FGC-feasibility of F. We fix deficient sets by performing a sequence of at most q augmentation iterations, where in each iteration we augment F with an edge-set $F' \subseteq E \setminus F$ such that $\delta(R) \cap F' \neq \emptyset$ for each $R \in \mathcal{C}$. We compute the desired F' via a reduction to the weighted set cover problem. First, let us state an upper bound on $|\mathcal{C}|$. For every $R \in \mathcal{C}$, observe that $\delta(R)$ is a $(1+\alpha)$-approximate min-cut in H, because H is p-edge connected (i.e., the size of a min-cut of H is $\geq p$). By Karger's bound [9], we have $|\mathcal{C}| \leq n^{2\alpha+2}$. In fact, with probability at least $1 - 1/n$, we can explicitly compute $\mathcal{C}$ in time polynomial in n^{α}. For simplicity, assume that we have explicit access to $\mathcal{C}$. Consider an instance of the weighted set cover problem where we want to cover elements of $\mathcal{C}$ by using sets of the form $\{\mathcal{R}_e\}_{e \in E \setminus F}$, where $\mathcal{R}_e := \{R \in \mathcal{C} : e \in \delta(R)\}$, and the weight of $\mathcal{R}_e$ is c_e. (Informally speaking, we have a ground-set of "points" that correspond to elements of $\mathcal{C}$, i.e., the deficient sets, we have a weighted set $\mathcal{R}_e$ corresponding to each edge $e \in E \setminus F$, and the goal is to pick a min-weight family of sets $\mathcal{R}_e$ whose union contains all the "points".) Since $F \cup F^*$ is feasible for the given (p, q)-FGC-instance, $\{\mathcal{R}_e\}_{e \in F^* \setminus F}$ is a feasible solution to the set-cover instance with cost at most $c(F^*)$. The well-known greedy algorithm for weighted set cover (see Theorem 13.3 in [16]) finds an $F' \subseteq E \setminus F$ satisfying $\delta(R) \cap F' \neq \emptyset$ for all $R \in \mathcal{C}$ and $c(F') \leq O(\alpha \log n) c(F^*)$. We augment F to $F \cup F'$ and discard the sets $R \in \mathcal{C}$ that are no longer deficient w.r.t. the augmented F. We repeatedly apply such augmenting iterations until $\mathcal{C}$ is empty. There are at most q such iterations, because each iteration increases the cardinality of $\delta(R) \cap F$ by one or more for each $R \in \mathcal{C}$. We summarize this discussion by the next claim.

▷ Claim. There is a randomized polynomial-time $O(q \log n)$-approximation algorithm for the special case of (p, q)-FGC where $q/p \leq O(1)$.

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. [6] and Chakrabarty et al. [3]. A well-studied problem in this area that is relevant to us is the Capacitated k-Connected Subgraph problem, see [3]. We denote this problem by Cap-k-ECSS. Formally, in an instance of this problem, we have an undirected multigraph $G = (V, E)$, nonnegative integer edge-capacities $\{u_e\}_{e \in E}$, nonnegative edge-costs $\{c_e\}_{e \in E}$, 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 $\sum_{e \in \delta(R) \cap F} u_e \geq k$, and $c(F)$ is minimized. Let n and m denote the number of vertices and edges of G, respectively. For this problem, Goemans et al. [6] give a $\min(2k, m)$-approximation algorithm, and Chakrabarty et al. [3] give a randomized $O(\log n)$-approximation algorithm.

In general, (p, q)-FGC and Cap-k-ECSS models are incomparable (see below for more details), however, when $p = 1$ or $q \leq 1$ holds, then (p, q)-FGC can be cast as an instance of the Cap-k-ECSS problem. The usual k-ECSS problem corresponds to the Cap-k-ECSS problem with unit edge capacities. The FGC problem corresponds to the Cap-2-ECSS problem, where safe edges have capacity 2 and unsafe edges have capacity 1. More generally, $(1, k)$-FGC corresponds to the Cap-$(k+1)$-ECSS problem, where safe edges have capacity $k+1$ and unsafe edges have capacity 1, and $(k, 1)$-FGC corresponds to the Cap-$(k(k+1))$-ECSS problem where safe edges have capacity $k+1$ and unsafe edges have capacity k. We remark that the most general models of (p, q)-FGC and Cap-k-ECSS are incomparable. In particular, it is easy to see that the (p, q)-FGC problem is not the same as the Cap-$(p(p+q))$-ECSS problem where safe edges have a capacity of $p+q$ and unsafe edges have a capacity of p. For instance, take $p = 2$ and $q = 3$: a cut with one safe edge (of capacity 5) and three unsafe edges (each with capacity 2) has total capacity $11 \geq p(p+q)$, but such a cut is deficient in the $(2,3)$-FGC model.

Our Contributions. We mention the main contributions of this work along with a brief overview of our results and techniques.

Our first result is a simple reduction from FGC to the well-known minimum-cost 2-arborescence problem that achieves an approximation guarantee of two. This result matches the current best approximation guarantee known for the 2ECSS problem, and improves on the 2.527-approximation algorithm of [2]. At a high level, our result is based on a straightforward extension of the 2-approximation algorithm of Khuller and Vishkin [10] for the 2ECSS problem. (In fact, Khuller and Vishkin [10] give a simple reduction from the k-ECSS problem to the problem of computing a minimum-cost k-arborescence in a digraph that achieves an approximation guarantee of two.)

▶ **Theorem 1.** *There is a* 2*-approximation algorithm for* FGC*.*

The following result generalizes Theorem 1 to the $(1,k)$-FGC problem, where we want to find a min-cost spanning subgraph that remains connected against the failure of any set of at most k unsafe edges.

▶ **Theorem 2.** *There is a* $(k+1)$*-approximation algorithm for* $(1,k)$-FGC*.*

Our proof of Theorem 2 is based on a reduction from $(1,k)$-FGC to the minimum-cost $(k+1)$-arborescence problem (see [13], Chapters 52 and 53). We lose a factor of $k+1$ in this reduction.

In Section 3, we consider the unweighted version of FGC, where each edge has unit cost. We design improved approximation algorithms for this special case.

▶ **Theorem 3.** *There is a* $\frac{16}{11}$*-approximation algorithm for unweighted* FGC*.*

In Section 4, we consider the $(k,1)$-FGC problem, where we seek a min-cost spanning subgraph that is k-edge connected against failure of at most one unsafe edge. Our main contribution here is the following.

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

Our algorithm in Theorem 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 k-ECSS problem, for which several 2-approximation algorithms are known. Let $H = (V,F)$ be the k-edge connected spanning subgraph 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 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 F on all deficient cuts. It turns out that this cut-augmentation problem can be formulated as the f-connectivity problem for an uncrossable function f (to be defined in Section 4). Williamson, Goemans, Mihail and Vazirani [17] present a 2-approximation algorithm for the latter problem.

Lastly, in Section 5, 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 5 is the following.

▶ **Theorem 5.** *There is a* $\min(k, 2u_{max})$*-approximation algorithm for the* Cap-k-ECSS *problem.*

Similar to Theorems 1 and 2, our proof of Theorem 5 is based on a reduction from the Cap-k-ECSS problem to the minimum-cost k-arborescence problem. The factor m in the $\min(2k, m)$ approximation guarantee of Goemans et al. comes from the fact that a simple greedy strategy yields an m-approximation for the Cap-k-ECSS problem. Assuming $\min(k, 2u_{max}) \leq m$, our result has a better dependence on k, and, in fact, for the standard case of $u_{min} = 1$, $u_{max} = k \ll m$, no previous result achieves an approximation guarantee of k (to the best of our knowledge). Our result above is incomparable to the result in [3]: our approximation guarantee 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 5 provides the following approximation guarantees for special cases of (p, q)-FGC:

(i) for $(1, 1)$-FGC, $k = u_{max} = 2$, so Theorem 5 gives a 2-approximation (same as Theorem 1);

(ii) for $(1, q)$-FGC, $k = u_{max} = q + 1$, so Theorem 5 gives a $(q + 1)$-approximation (same as Theorem 2); and

(iii) for $(p, 1)$-FGC with $p > 1$, $k = p(p + 1)$ and $u_{max} = p + 1$, so Theorem 5 gives a $2(p + 1)$-approximation (this is weaker than the 4-approximation given by Theorem 4).

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) follows as a special case. Recall that in an instance of $(1, k)$-FGC we have an undirected multigraph $G = (V, E)$ (with no self loops), a partition of $E = \mathcal{S} \sqcup \mathcal{U}$ into safe and unsafe edges, and nonnegative edge-costs $\{c_e\}_{e \in E}$. Our objective is to find a minimum-cost edge-set $F \subseteq E$ such that the subgraph (V, F) remains connected against failure of any k unsafe edges.

For a subgraph H of G and a nonempty vertex-set $S \subsetneq V$, we use $\delta_H(S)$ to denote the set of edges in H with exactly one endpoint in S, i.e., $\delta_H(S) := \{e = uv \in E(H) : |\{u, v\} \cap S| = 1\}$. We drop the subscript H when the underlying graph is clear from the context. The following characterization of $(1, k)$-FGC solutions is straightforward.

▶ **Proposition 6.** *F 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.*

For the rest of the paper, we assume that the given instance of $(1, k)$-FGC is feasible: this can be checked by computing a (global) minimum cut in G where we assign a capacity of $k+1$ to safe edges and a capacity of 1 to unsafe edges. As mentioned before, our algorithm for $(1, k)$-FGC is based on a reduction to the minimum-cost r-out $(k + 1)$-arborescence problem. We state a few standard results on arborescences. Let $D = (W, A)$ be a digraph and $\{c'_a\}_{a \in A}$ 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 7** (r-out arborescence)**.** *An r-out 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-out arborescence is a directed spanning tree rooted out of r. More generally, an r-out k-arborescence is a union of k arc-disjoint r-out arborescences.

▶ **Definition 8** (r-out k-arborescence). *For a positive integer k, a subgraph (W, T) is an r-out k-arborescence if T can be partitioned into k arc-disjoint r-out arborescences.*

The following results on existence of arborescences and the corresponding optimization problem will be useful to us.

▶ **Theorem 9** ([13], Chapter 53.8). *Let $D = (W, A)$ be a digraph, $r \in W$ be a root vertex, and k be a positive integer. Then, D contains an r-out k-arborescence if and only if $|\delta^{\text{in}}_D(S)| \geq k$ for any nonempty vertex-set $S \subseteq V \setminus \{r\}$.*

▶ **Theorem 10** ([13], Theorem 53.10). *In strongly polynomial time, we can obtain an optimal solution to the minimum c'-cost r-out k-arborescence problem on D, or conclude that there is no r-out k-arborescence in D.*

The following claim is useful in our analysis.

▷ Claim 11. Let (W, T) be an r-out 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 endpoint at u and the other endpoint at v (counting multiplicities) is at most k.

Proof. Since an r-out k-arborescence is a union of k arc-disjoint r-out 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. ◁

In our proofs we move from undirected graphs to their directed counterparts by bidirecting edges. We formalize this notion.

▶ **Definition 12** (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 $(1, k)$-FGC solution F can be used to obtain an r-out $(k + 1)$-arborescence (in an appropriate digraph) of cost at most $(k + 1)c(F)$.

▶ **Lemma 13.** *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-out $(k + 1)$-arborescence in D with cost at most $(k + 1)c(F)$.*

Proof. Let (V, T) be a minimum-cost r-out $(k + 1)$-arborescence in D. First, we argue that T is well-defined. By Theorem 9, 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 6). 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 11 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 13 naturally suggests a reduction from $(1, k)$-FGC to the minimum-cost r-out $(k+1)$-arborescence problem. We prove the main theorem of this section.

Proof of Theorem 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 c-cost r-out $(k+1)$-arborescence in D. By Lemma 13, $c(T) \le (k+1)c(F^*)$, where F^* denotes an optimal $(1, k)$-FGC solution to the given instance.

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 $c(F) \le c(T)$. 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-out $(k+1)$-arborescence, by Theorem 9 we have $|\delta^{\text{in}}_T(S)| \ge k+1$. 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 both orientations of an edge cannot appear in $\delta^{\text{in}}_D(S)$, we get that $|F \cap \mathcal{U} \cap \delta(S)| \ge k+1$. By Proposition 6, F is a feasible solution for the given instance of $(1, k)$-FGC with $c(F) \le (k+1)\text{OPT}$. ◀

3 Unweighted 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 3); to the best of our knowledge, this is the first result that provides a better than $\frac{3}{2}$ approximation for 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 unweighted 2ECSS problem: this implies a $\frac{5}{3}$-approximation algorithm for unweighted FGC by using the result of Sebö and Vygen [14]. 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 no such algorithm can obtain an approximation factor better than $\frac{3}{2}$. Our main result in this section is the following.

▶ **Theorem 14.** *Suppose that there is an α-approximation algorithm for the unweighted* 2ECSS *problem. Then, there is a $\frac{4\alpha}{2\alpha+1}$-approximation algorithm for unweighted* FGC*.*

Theorem 3 follows from the above theorem by using the $\frac{4}{3}$-approximation algorithm of Sebö and Vygen [14] for the unweighted 2ECSS problem. Before delving into the proof of Theorem 14, 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 multigraph with no self-loops and let $\{c'_e\}_{e \in E'}$ be nonnegative costs on the edges.

▶ **Definition 15** (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.

▶ **Theorem 16** ([13], Theorem 29.1)**.** *In strongly polynomial time, we can obtain a minimum c'-cost W-join, or conclude that there is no W-join in G'.*

The W-join polytope is the convex hull of the incidence vectors of W-joins. Its dominant has a simple linear description.

▶ **Theorem 17** ([13], 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 multigraph $G = (V, E = \mathcal{S} \cup \mathcal{U})$ with a specified partition of E into safe and unsafe edges. We will assume that G is connected and has no unsafe bridges, since otherwise the instance is infeasible. Let F^* denote an optimal solution. Suppose that we have access to an α-approximation algorithm for the 2ECSS problem. We give two algorithms for obtaining two candidate solutions to the given instance. We then argue that the cheaper of these two solutions is a $\frac{4\alpha}{2\alpha+1}$-approximate solution.

Join-based Algorithm for Unweighted FGC. Let T be a spanning tree in G that maximizes the number of 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) collection of unsafe edges in T. Let $G' = (V', E')$ denote the graph obtained from G by contracting (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 and T' is a spanning tree of G'. Let W' denote the (nonempty) set of odd degree vertices in the subgraph (V', T'). Let $J' \subseteq E'$ be a minimum-cardinality W'-join in G', which we can compute in polynomial time by using Theorem 16. By our choice, the subgraph $(V', T' \sqcup J')$ is connected and Eulerian, so it is 2-edge connected in G'. Consider the multiset $F_1 = T \sqcup J'$ 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 S 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 $|F_1| \leq |T| + |J'|$. The following claim gives a bound on the quality of our first solution.

▷ Claim 18. 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 so that $T \setminus T'$ is a maximal safe forest in G, and we obtained G' by contracting connected components in $(V, T \setminus T')$. By our assumption that G has no unsafe bridges, we have that G' is 2-edge connected and consists of only unsafe edges. 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 S' be an arbitrary cut in G' and let S be the unique cut in G that gives rise to 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 Theorem 17, 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 $|T| \leq |F^*|$:

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

The above claim shows that the size of F_1 can be charged to a certain combination of the number of safe and unsafe edges in F^*. Our second algorithm uses the α-approximation for the 2ECSS problem as a subroutine. The solution returned by this algorithm has the property that its size complements that of F_1.

2ECSS-based Algorithm for Unweighted FGC. Consider the multigraph 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 unweighted 2ECSS 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 19. 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 14. 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 2ECSS problem. We show that the smaller of F_1 and F_2 is a $\frac{4\alpha}{2\alpha+1}$-approximate solution for the given unweighted FGC-instance. By Claims 18 and 19:

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

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 approximation guarantee of $\frac{3}{2}$ or more.

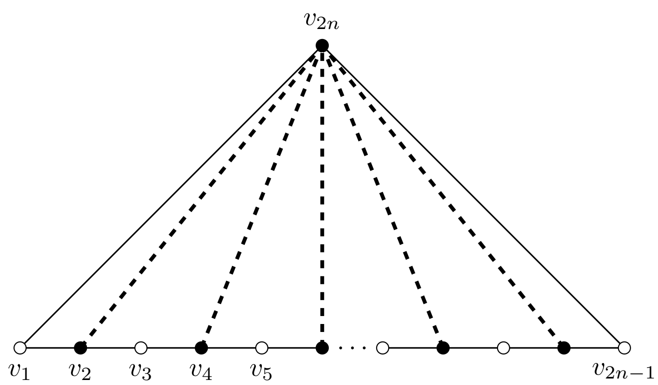

Figure 1 In this instance we have a graph on $2n$ vertices. The set of unsafe edges, shown using solid lines, forms a Hamiltonian cycle. For each $i = 1, \ldots, n-1$, there is a safe edge, shown using a thick dashed line, between v_{2i} and v_{2n}. The solution consisting of all unsafe edges is feasible,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

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 4). Recall that in an instance of $(k,1)$-FGC, we have a multigraph $G=(V,E=\mathcal{S}\cup\mathcal{U})$ with a partition of the edge-set into safe and unsafe edges, nonnegative edge-costs $\{c_e\}_{e\in E}$, 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 yields a better approximation guarantee than Theorem 4. Let F^* denote an optimal solution to the given instance. The following characterization of $(k,1)$-FGC solutions is straightforward.

▶ **Proposition 20.** *F 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.*

The above proposition suggests a two-stage strategy for $(k,1)$-FGC. Suppose that in the first stage we compute a cheap k-edge connected spanning subgraph $H_1=(V,F_1)$ of G without making any distinction between safe and unsafe edges. For any nonempty cut $S\subsetneq V$, we have $\delta_{H_1}(S)\geq k$, so by Proposition 20, the only hindrance to the $(k,1)$-FGC feasibility of F_1 are k-cuts in H_1 that contain at least one unsafe edge. We call such cuts *deficient*. The subproblem remaining for the second stage is an augmentation problem for these *deficient* cuts, which is special case of the (minimum-cost) f-connectivity problem.

In the *f-connectivity* problem we have an undirected multigraph $G'=(V',E')$, nonnegative edge-costs $\{c'_e\}_{e\in E'}$, and a cut-requirement function $f:2^{V'}\rightarrow\{0,1\}$ 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 for the f-connectivity problem 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 cut S with $f(S)=1$ there is at least one F-edge in this cut. The objective is to find a feasible $F\subseteq E'$ that minimizes $c(F)$. The f-connectivity problem can be modeled as an integer program whose linear relaxation (P) is stated below. For each edge $e\in E'$ the LP has a nonnegative variable x_e that models the extent to which the edge e is picked by the solution.

$$\begin{aligned} &\min \sum_{e\in E'} c'_e x_e && && \text{(P)}\\ &\text{subject to } x(\delta_{G'}(S))\geq 1 && \forall\, S\subseteq V' \text{ s.t. } f(S)=1\\ &\qquad\qquad x_e\geq 0 && \forall\, e\in E'. \end{aligned}$$

The f-connectivity problem has received a lot of attention in Combinatorial Optimization since it captures many well-known network design problems. In particular, it captures the generalized Steiner network problem. Williamson et al. [17] gave a primal-dual framework to obtain approximation algorithms for the f-connectivity problems when f is a proper function, and more generally, when f is an uncrossable function (also see the book chapter by Geomans and Williamson [7] for an excellent survey on primal-dual algorithms for network design problems).

▶ **Definition 21** (Uncrossable function)**.** *A function $f:2^{V'}\rightarrow\{0,1\}$ 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 algorithm, the primal-dual algorithm of [17] gives a 2-approximation for the f-connectivity problem with an uncrossable function f. There is no explicit result in [17] that can be quoted verbatim and applied for our purposes, so we reference the most relevant lemma from their work.

▶ **Definition 22** (Minimal violated sets). *Let $f : 2^{V'} \to \{0, 1\}$ be a cut-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 S is inclusion-wise minimal among all violated sets.*

▶ **Theorem 23** ([17], Lemma 2.1). *Let $f : 2^{V'} \to \{0, 1\}$ be 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. We can compute a 2-approximate solution to the f-connectivity problem in polynomial time.*

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

Description of Our 4-Approximation Algorithm for $(k, 1)$-FGC. Our algorithm runs in two stages. In the first stage, we compute a 2-approximate k-edge connected spanning subgraph $H_1 = (V, F_1)$ of G without making any distinction between safe and unsafe edges; since F^* is k-edge connected, the k-ECSS instance is feasible. This can be done using Jain's iterative rounding algorithm [8]. Next, we compute the collection $\mathcal{C} = \{S \subsetneq V : |\delta(S) \cap F_1| = k\}$ of all (minimum) k-cuts in H_1. Consider the cut-requirement function $f : 2^V \to \{0, 1\}$ where $f(S)$ is 1 if and only if $S \in \mathcal{C}$ and $F_1 \cap \delta(S) \cap \mathcal{U} \neq \emptyset$. Consider an instance of the f-connectivity problem for the graph $G' := G - F_1$ with edge-costs $\{c_e\}_{e \in E \setminus F_1}$; note that $F^* \setminus F_1$ is feasible to this f-connectivity instance. In the second stage, we use Theorem 23 to compute a 2-approximate solution $F_2 \subseteq E \setminus F_1$ for this f-connectivity instance. We return the solution $F = F_1 \sqcup F_2$.

To prove Theorem 4, we need to argue the following: (i) f is uncrossable; (ii) we can compute minimal violated sets (w.r.t f and any $F' \subseteq E \setminus F_1$) in polynomial time; (iii) F is a feasible $(k, 1)$-FGC solution; (iv) $c(F) \leq 4c(F^*)$; (v) the whole algorithm runs in polynomial time. We defer the proofs of (i) and (ii) to the end of this section. Assuming that they are true, (v) follows from Theorem 23. The following lemma covers (iii) and (iv).

▶ **Lemma 24.** *The edge-set F 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, F is a subgraph of G. We use the characterization of feasible solutions given by Proposition 20. Let $S \subsetneq V$ be an arbitrary nonempty cut. 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 $|F \cap \delta(S)| \geq k + 1$, and we are done. Otherwise, S is a k-cut in 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, and we are done. Otherwise, by definition, $f(S) = 1$. Next, by feasibility of F_2 for f-connectivity, we have $F_2 \cap \delta(S) \neq \emptyset$. So, $|F \cap \delta(S)| = |F_1 \cap \delta(S)| + |F_2 \cap \delta(S)| \geq k + 1$, and we are done.

We show that F is 4-approximate by arguing that $c(F_1)$ and $c(F_2)$ are bounded by $2c(F^*)$. The bound on $c(F_1)$ is immediate from the fact that F^* is feasible for the k-ECSS instance considered in Stage 1. Next, by feasibility of $F^* \setminus F_1$ for the f-connectivity instance, we have $c(F_2) \leq 2c(F^* \setminus F_1) \leq 2c(F^*)$, so we are done. ◀

▷ Claim 25. For any $F' \subseteq E \setminus F_1$, we can compute all minimal violated sets w.r.t. f and F'.

Proof. Since a graph on n vertices has at most $O(n^2)$ min-cuts [9], we have $|\mathcal{C}| = O(|V|^2)$. Using standard network flow algorithms, we can compute $\mathcal{C}$ in polynomial time (for instance, see [4]). Since we have explicit access to $\mathcal{C}$, we have a value oracle for f. Fix some $F \subseteq E \setminus F_1$. Any violated set must have $f(S) = 1$, so there are at most $|\mathcal{C}|$ many violated sets. We can exhaustively go through all violated sets and find the minimal elements. ◁

Lastly, we show that f is uncrossable.

▶ **Lemma 26.** *f is uncrossable.*

Proof. We check if the two properties of an uncrossable function hold for f (recall Definition 21); $f(V) = 0$ is trivial. Symmetry of f follows from 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, in the subgraph $H_1 = (V, F_1)$, both A and B are (minimum) k-cuts with at least one unsafe edge on their respective boundaries. 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. By submodularity of the function $d(S) := |\delta_{H_1}(S)|$, we get:

$$|\delta_{H_1}(A \cap B)| = |\delta_{H_1}(A \cup B)| = |\delta_{H_1}(A \setminus B)| = |\delta_{H_1}(B \setminus A)| = k. \tag{1}$$

Furthermore, we also have:

$$F_1 \cap E(A \setminus B, B \setminus A) = \emptyset \quad \text{and} \quad F_1 \cap E(A \cap B, V \setminus (A \cup B)) = \emptyset, \tag{2}$$

where $E(S, T)$ denotes the set of edges in G with one endpoint in S and the other endpoint in T. We finish the proof by doing a case analysis on e_1 and e_2. By (2), exactly one of the following happens: (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) happens, then $f(A \setminus B) = f(A \cup B) = 1$. Otherwise, $f(A \cap B) = f(B \setminus A) = 1$. We do a similar analysis on e_2. Exactly one of the following happens: (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) happens, then $f(B \setminus A) = f(A \cup B) = 1$. Otherwise, $f(A \cap B) = f(A \setminus B) = 1$. 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 $f(A \setminus B) = f(B \setminus A) = 1$. ◀

5 The Capacitated k-Connected Subgraph Problem

In this section we consider the Cap-k-ECSS problem. We are given a multigraph $G = (V, E)$, nonnegative integer edge-capacities $\{u_e\}_e$, nonnegative edge-costs $\{c_e\}_e$, 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 $\sum_{e \in \delta(R) \cap F} u_e \geq k$, 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 instance is feasible. This can be verified in polynomial time by checking if G has any cut with capacity 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 5); our algorithm is based on a reduction to the min-cost 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-out k-arborescence (see Theorem 9). We use Theorem 10 on (D, c) to obtain a minimum-cost r-out k-arborescence T' in polynomial time. Let F' be the set of all edges $e \in E$ for which at least one of the corresponding $2u_e$ arcs in D appears in the optimal r-out k-arborescence T'.

▶ **Lemma 27.** *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 an arbitrary nonempty vertex-set that excludes the root vertex r. Since T' contains k arc-disjoint r-out arborescences, $|\delta^{\text{in}}_{T'}(R)| \geq k$. For each edge $e \in E$, at most u_e of the corresponding arcs in D can appear in the set of T'-arcs entering R. Thus, $\sum_{e \in \delta(R) \cap F'} u_e \geq |\delta^{\text{in}}_{T'}(R)| \geq k$, 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 corresponding $2u_e$ arcs appear in T', so we have $c(F') \leq c(T')$. ◀

We now prove Theorem 5 by showing that F' is the desired $\min(k, 2u_{max})$-approximate solution.

Proof of Theorem 5. Let $(G(V, E), u, c, k)$ be a feasible instance of the Cap-k-ECSS problem. Let $D = (V, A)$ be the digraph and T' be the r-out k-arborescence as constructed by our 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. Let $r \in V$ be the root vertex fixed by the algorithm. By feasibility of F^* (for the Cap-k-ECSS instance), we know that D^* contains an r-out k-arborescence. Let T^* denote an optimal r-out k-arborescence in D^*. Since D^* is a subgraph of D and T' is an optimal r-out k-arborescence in D, we have $c(T') \leq c(T^*)$. By Lemma 27, $c(F') \leq c(T')$, so to prove the theorem it suffices to argue that $c(T^*) \leq \min(t, 2u_{max})c(F^*)$. To this end, observe that for any edge $e \in F^*$ there can be at most $2u_e$ arcs in A^* by construction of D^*. Hence, $c(T^*) \leq c(A^*) \leq 2u_{max}c(F^*)$ holds. Next, by definition, T^* can be partitioned into k (arc-disjoint) r-out arborescences, each of which can use at most one of the $2u_e$ arcs corresponding to an edge e of G. It follows that for each edge $e \in F^*$ at most k of the corresponding $2u_e$ arcs can appear in T^*. Therefore, $c(T^*) \leq kc(F^*)$. This completes the proof. ◀

References

1. David Adjiashvili, Felix Hommelsheim, and Moritz 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. doi:10.1007/978-3-030-45771-6_2.
2. David Adjiashvili, Felix Hommelsheim, and Moritz Mühlenthaler. Flexible Graph Connectivity. *Mathematical Programming*, pages 1–33, 2021. doi:10.1007/s10107-021-01664-9.
3. Deeparnab Chakrabarty, Chandra Chekuri, Sanjeev Khanna, and Nitish Korula. Approximability of Capacitated Network Design. *Algorithmica*, 72(2):493–514, 2015. doi:10.1007/s00453-013-9862-4.

4 Lisa 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. doi:10.1006/jagm.1999.1039.

5 Harold N. Gabow, Michel X. Goemans, Éva Tardos, and David P. Williamson. Approximating the Smallest k-Edge Connected Spanning Subgraph by LP-rounding. *Networks*, 53(4):345–357, 2009. doi:10.1002/net.20289.

6 Michel X. Goemans, Andrew V. Goldberg, Serge A. Plotkin, David B. Shmoys, Éva Tardos, and David P. Williamson. Improved Approximation Algorithms for Network Design Problems. In *Proceedings of the 5th Symposium on Discrete Algorithms*, pages 223–232, 1994. doi:10.5555/314464.314497.

7 Michel X. Goemans and David 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. URL: https://math.mit.edu/~goemans/PAPERS/book-ch4.pdf.

8 Kamal Jain. A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. *Combinatorica*, 21(1):39–60, 2001. doi:10.1007/s004930170004.

9 David 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. doi:10.5555/313559.313605.

10 Samir Khuller and Uzi Vishkin. Biconnectivity Approximations and Graph Carvings. *Journal of the ACM*, 41(2):214–235, 1994. doi:10.1145/174652.174654.

11 Thomas L. Magnanti and Richard T. Wong. Network Design and Transportation Planning: Models and Algorithms. *Transportation Science*, 18(1):1–55, 1984. doi:10.1287/trsc.18.1.1.

12 Polina Rozenshtein, Aristides Gionis, B. Aditya Prakash, and Jilles Vreeken. Reconstructing an Epidemic Over Time. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1835–1844, 2016. doi:10.1145/2939672.2939865.

13 Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag Berlin Heidelberg, 2003.

14 András Sebö and Jens 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. doi:10.1007/s00493-014-2960-3.

15 Lawrence V. Snyder, Maria P. Scaparra, Mark S. Daskin, and Richard L. Church. *Planning for Disruptions in Supply Chain Networks*, pages 234–257. Institute for Operations Research and the Management Sciences (INFORMS), 2014. doi:10.1287/educ.1063.0025.

16 Vijay V. Vazirani. *Approximation Algorithms.* Springer Berlin Heidelberg, 2003. doi:10.1007/978-3-662-04565-7.

17 David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani. A Primal-Dual Approximation Algorithm for Generalized Steiner Network Problems. *Combinatorica*, 15(3):435–454, 1995. doi:10.1007/BF01299747.