Improved Approximation Algorithms for Flexible Graph Connectivity and Capacitated Network Design

Ishan Bansal* Joseph Cheriyan† Sanjeev Khanna‡ Miles Simmons§

September 14, 2025

Abstract

We present improved approximation algorithms for some problems in the related areas of Flexible Graph Connectivity and Capacitated Network Design. In the (p, q)-Flexible Graph Connectivity problem, denoted (p, q)-FGC, the input is a graph $G(V, E)$ where E is partitioned into *safe* and *unsafe* edges, and the goal is to find a minimum cost set of edges F such that the subgraph $G'(V, F)$ remains p-edge connected upon removal of any q unsafe edges from F. In the related Cap-k-ECSS problem, we are given a graph $G(V, E)$ whose edges have arbitrary integer capacities, and the goal is to find a minimum cost subset of edges F such that the graph $G'(V, F)$ is k-edge connected.

We obtain a 7-approximation algorithm for the $(1, q)$-FGC problem that improves upon the previous best $(q + 1)$-approximation. We also give an $O(\log k)$-approximation algorithm for the Cap-k-ECSS problem, improving upon the previous best $O(\log n)$-approximation whenever $k = o(n)$. Both these results are obtained by using natural LP relaxations strengthened with the knapsack-cover inequalities, and then during the rounding process utilizing an $O(1)$-approximation algorithm for the problem of covering small cuts. We also show that the the problem of covering small cuts inherently arises in another variant of (p, q)-FGC. Specifically, we show $O(1)$-approximate reductions between the $(2, q)$-FGC problem and the 2-Cover Small Cuts problem where each small cut needs to be covered twice.

1 Introduction

We study some problems in the related areas of Flexible Graph Connectivity and Capacitated Network Design.

Flexible Graph Connectivity and the (p, q)-FGC problem

Adjiashvili, Hommelsheim and Mühlenthaler [1] introduced the model of Flexible Graph Connectivity that we denote by FGC as a way to model network design problems where edges have non-uniform reliability. Boyd, Cheriyan, Haddadan and Ibrahimpur [5] introduced a generalization of FGC, called (p, q)-Flexible Graph Connectivity problem, denoted (p, q)-FGC, where p is an integer denoting *connectivity* requirement, and q is an integer denoting the *robustness* requirement. An instance of (p, q)-FGC consists of an undirected graph $G = (V, E)$, where E is partitioned into a set of safe edges $\mathcal{S}$ (edges that never fail) and a set of unsafe edges $\mathcal{U}$ (edges that may fail), and nonnegative edge-costs $c \in \mathbb{Q}^E_{\geq 0}$. A subset $F \subseteq E$ of edges is feasible for the (p, q)-FGC problem if for any set F' consisting of at most q unsafe edges, the subgraph $(V, F - F')$ remains p-edge connected. The objective is to find a feasible solution F that minimizes $c(F) = \sum_{e \in F} c_e$.

Boyd et al. [5] presented a 4-approximation algorithm for $(p, 1)$-FGC based on the primal-dual method of Williamson, Goemans, Mihail & Vazirani (WGMV) [19], and a $(q + 1)$-approximation algorithm for $(1, q)$-FGC; moreover, they gave an $O(q \log n)$-approximation algorithm for (general) (p, q)-FGC. Subsequently, numerous results and approximation algorithms have been presented by Bansal, Bansal et al., Chekuri & Jain, Nutov, etc., see [2, 4, 8, 17].

*ib332@cornell.edu. Amazon, Bellevue, WA, USA. This work is external and does not relate to the position at Amazon.

†jcheriyan@uwaterloo.ca. Department of Combinatorics & Optimization, University of Waterloo, Canada.

‡sanjeev@cis.upenn.edu. University of Pennsylvania, Philadelphia, PA, USA.

§mjsimmons@uwaterloo.ca. Department of Combinatorics & Optimization, University of Waterloo, Canada.

Capacitated Network Design and the Cap-k-ECSS problem

The Cap-k-ECSS problem is as follows: Given an undirected graph $G = (V, E)$ with edge costs $c \in \mathbb{Q}^E_{\geq 0}$ and edge capacities $u \in \mathbb{Z}^E_{\geq 0}$, find a minimum-cost subset of the edges $F \subseteq E$ such that the capacity of any cut in (V, F) is at least k. Let u_{min} (respectively, u_{max}) denote the minimum (respectively, maximum) capacity of an edge in E, and assume (w.l.o.g.) that $u_{max} \leq k$.

The best known approximation ratio for Cap-k-ECSS is $\min(O(\log |V|), k, 2u_{max}, 5 \cdot \lceil k/u_{min} \rceil)$, due to Chakrabarty et al., Goemans et al., & Bansal [7, 12, 2].

For a graph $G = (V, E)$ and a set of nodes $S \subseteq V$, the *cut of* S, denoted by $\delta(S)$, refers to the set of edges that have exactly one end-node in S. We call a cut $\delta(S)$ *nontrivial* if S is a nonempty, proper subset of V, that is, if $\emptyset \neq S \subsetneq V$.

The following integer program formulates the Cap-k-ECSS problem.

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e & & \text{(IP: CapkECSS)} \\
\text{s.t.} \quad & \sum_{e \in E \cap \delta(S)} u_e x_e \geq k & \forall \ \emptyset \subsetneq S \subsetneq V & \\
& x_e \in \{0, 1\} & \forall \ e \in E &
\end{aligned}
$$

The LP (linear programming) relaxation of the above integer program is obtained by replacing $x_e \in \{0, 1\}$ by $0 \leq x_e \leq 1$, $\forall e \in E$. The following well-known example shows that the LP relaxation has integrality ratio $\Omega(k)$.

The graph G consists of a pair of nodes, and a pair of parallel edges e_1, e_2 between the two nodes. Edge e_1 has cost zero and capacity $k - 1$, and edge e_2 has cost one and capacity k. A feasible solution of the integer program has cost one (since e_2 must be picked). On the other hand, a feasible solution x to the LP relaxation of cost $1/k$ is given by $x_{e_1} = 1, x_{e_2} = 1/k$.

Knapsack-Cover Inequalities (KCI) for Capacitated Network Design

The LP relaxation of (IP: CapkECSS) can be strengthened using the Knapsack-Cover Inequalities (KCI). For any non-trivial cut S and a subset of the edges $A \subseteq E$, the following is a valid inequality for all integer solutions of (IP: CapkECSS).

$$
\sum_{e \in E \cap \delta(S) - A} u_e(A, S) x_e \geq D(A, S)
$$

where $D(A, S) = \max\{0, k - \sum_{e \in \delta(S) \cap A} u_e\}$ and $u_e(A, S) = \min\{u_e, D(A, S)\}$. (By plugging in $A = \emptyset$, we get the constraint $\sum_{e \in E \cap \delta(S)} u_e x_e \geq k$, which is a constraint of (IP: CapkECSS).)

We add these inequalities to (IP: CapkECSS) to obtain the following LP relaxation of the Cap-k-ECSS problem.

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e & & \text{(KCLP: CapkECSS)} \\
\text{s.t.} \quad & \sum_{e \in E \cap \delta(S) - A} u_e(A, S) x_e \geq D(A, S) & \forall \ \emptyset \subsetneq S \subsetneq V, A \subseteq E & \\
& 0 \leq x_e \leq 1 & \forall \ e \in E &
\end{aligned}
$$

Observe that this LP has a number of constraints that is exponential in the size of the input instance (of Cap-k-ECSS). By following the cut-and-round approach employed by Carr, Fleischer, Leung & Phillips [6], one can round this LP in polynomial time to an approximately optimal integer solution via the ellipsoid method by designing an efficient pseudo-separation subroutine. See sections 2, 3, 4 for details.

LP relaxations for (p, q)-FGC

The following linear program gives a lower bound on the optimal solution value for (p, q)-FGC. Such LP relaxations are discussed in [5] and [8, Section 2].

To motivate the LP relaxation, consider an auxiliary capacitated graph that has the same set of nodes and the same set of edges as the graph of the (p, q)-FGC instance. Assign a capacity of $(p + q)$ to each safe edge and a capacity of p to each unsafe edge. Let $k = p(p + q)$ and view the capacitated graph as an instance of the Cap-k-ECSS problem. In general, observe that a feasible solution of the (p, q)-FGC instance corresponds to a feasible solution of the Cap-k-ECSS instance, but not vice-versa. (When either $p = 1$ or $q = 1$, then a feasible solution of the Cap-k-ECSS instance corresponds to a feasible solution of the (p, q)-FGC instance.) Our LP relaxation can be viewed as the natural "cut covering" LP relaxation of the Cap-k-ECSS problem. Each safe edge e has a variable x_e, each unsafe edge f has a variable y_f, and for any set of nodes S, we use the notation $x(\delta(S)) \equiv \sum\{x_e \mid e \in \delta(S) \cap \mathcal{S}\}$, $y(\delta(S)) \equiv \sum\{y_f \mid f \in \delta(S) \cap \mathcal{U}\}$.

$$
\begin{aligned}
\mathrm{LP_{opt}} := \quad & \min \sum_{e\in\mathcal{S}} c_e x_e + \sum_{f\in\mathcal{U}} c_f y_f && && (1)\\
\text{s.t.} \quad & (p)y(\delta(S)) + (p+q)x(\delta(S)) \geq p(p+q) && \forall \quad \emptyset \subsetneq S \subsetneq V && (2)\\
& 0 \leq x_e \leq 1 && \forall e \in \mathcal{S} && (3)\\
& 0 \leq y_f \leq 1 && \forall f \in \mathcal{U} && (4)
\end{aligned}
$$

The f-connectivity problem and Jain's iterative rounding algorithm

Most connectivity augmentation problems can be formulated in a general framework called f-connectivity. In this problem, we are given an undirected graph $G = (V, E)$ on n nodes with nonnegative costs $c \in \mathbb{Q}^E_{\geq 0}$ on the edges and a requirement function $f : 2^V \to \mathbb{Z}_{\geq 0}$ on subsets of nodes. We are interested in finding an edge-set $J \subseteq E$ with minimum cost $c(J) := \sum_{e\in J} c_e$ such that for all cuts $\delta(S)$, $S \subseteq V$, we have $|\delta(S) \cap J| \geq f(S)$. This problem can be formulated as the following integer program where the binary variable x_e models the inclusion of the edge e in J:

$$
\begin{aligned}
\min \quad & \sum_{e\in E} c_e x_e && && \text{(IP: } f\text{-connectivity)}\\
\text{subject to:} \quad & x(\delta(S)) \geq f(S) && \forall\ S \subseteq V &&\\
& x_e \in \{0, 1\} && \forall\ e \in E. &&
\end{aligned}
$$

Assuming that the function f is weakly supermodular, ~~(and integral)~~ integral, and has a positive value for some $S \subset V$, Jain [14] presented a 2-approximation algorithm for the f-connectivity problem. Recently, Dinitz et al. [10, 11] discussed extensions of Jain's methods to the setting of locally weakly supermodular functions and weakly $\mathcal{F}$-supermodular functions. In section 3, we use the latter notion.

Let $\mathcal{F}$ be a family of subsets of V, such that $\emptyset, V \notin \mathcal{F}$. A function f is called *weakly $\mathcal{F}$-supermodular* if, $f(S) = 0$, $\forall S \notin \mathcal{F}$, and for all $A, B \in \mathcal{F}$, at least one of the following holds:

- $A - B, B - A \in \mathcal{F}$ and $f(A) + f(B) \leq f(A - B) + f(B - A)$
- $A \cap B, A \cup B \in \mathcal{F}$ and $f(A) + f(B) \leq f(A \cap B) + f(A \cup B)$

Remark. *The reader can follow our presentation independently of [11]; nevertheless, we mention that [11] has a few typos, and there is confusion pertaining to the functions f and $f_{\mathcal{F}}$. Our presentation does not use the notion of $f_{\mathcal{F}}$ and our definition of a weakly $\mathcal{F}$-supermodular function f adds the condition $f(S) = 0$, $\forall S \notin \mathcal{F}$.*

The following result is an extension of [14, Theorem 2.5].

Proposition 1. *Let G be a graph and let $x \in \{0, 1\}^{E(G)}$ denote a subgraph of G. Let $f : 2^{V(G)} \to \mathbb{Z}$ be a weakly $\mathcal{F}$-supermodular function. Then $f(S) - |\delta_x(S)|$ is also a weakly $\mathcal{F}$-supermodular function.*

Moreover, [14, Lemma 4.1] holds when the function f is a weakly $\mathcal{F}$-supermodular function. Let us call a node set S *tight* if $x(\delta(S)) = f(S)$, and let χ^S denote the incidence vector of $\delta(S)$. [14, Lemma 4.1] states that if $A, B \subseteq V(G)$ are tight sets, then either $A - B, B - A$ are tight sets and $\chi^A + \chi^B = \chi^{A-B} + \chi^{B-A}$, or $A \cap B, A \cup B$ are tight sets and $\chi^A + \chi^B = \chi^{A \cap B} + \chi^{A \cup B}$.

The other relevant lemmas and theorems of [14] continue to hold for a weakly $\mathcal{F}$-supermodular function f; in particular, [14, Theorem 3.1] holds; it states that for any basic solution x of the LP relaxation of the f-connectivity problem, there is an edge e such that $x_e \geq 1/2$.

In section 3, we define requirement functions f of the form

$$f(S) = \begin{cases} k, & \text{if } \emptyset \neq S \subsetneq V \text{ and } |\delta(S) \cap E_0| \leq 1 \\ 0, & \text{otherwise,} \end{cases}$$

where k is a positive integer and E_0 is a subset of $E(G)$. Let $\mathcal{F}$ be the family of nonempty, proper subsets S of $V(G)$ such that $|\delta(S) \cap E_0| \leq 1$. For any two sets $A, B \in \mathcal{F}$, either $A - B, B - A \in \mathcal{F}$ or $A \cap B, A \cup B \in \mathcal{F}$; this can be proved via case analysis; for example, see the last part of the proof of [5, Lemma 4.3]. Hence, it follows that the above requirement function f is weakly $\mathcal{F}$-supermodular. This gives the next result.

Proposition 2. *Jain's iterative rounding algorithm finds a 2-approximate solution to an f-connectivity problem and a pre-selected edge set E_0 such that f has the form* $f(S) = \begin{cases} k, & \text{if } \emptyset \neq S \subsetneq V \text{ and } |\delta(S) \cap E_0| \leq 1 \\ 0, & \text{otherwise.} \end{cases}$

The Cover Small Cuts problem

We follow the notation from [3, Section 1.3]. In an instance of the Cover Small Cuts problem, we are given an undirected capacitated graph $G = (V, E)$ with edge-capacities $u \in \mathbb{Q}^E_{\geq 0}$, a set of links $L \subseteq \binom{V}{2}$ with costs $c \in \mathbb{Q}^L_{\geq 0}$, and a threshold $\lambda \in \mathbb{Q}_{\geq 0}$. A subset $F \subseteq L$ of links is said to *cover* a node-set S if there exists a link $e \in F$ with exactly one end-node in S. The objective is to find a minimum-cost $F \subseteq L$ that covers each non-empty $S \subsetneq V$ with $u(\delta_E(S)) < \lambda$.

Let $\mathcal{C} = \{\emptyset \neq S \subsetneq V : u(\delta_E(S)) < \lambda\}$. Then we have the following covering LP relaxation of the problem.

$$\begin{aligned} \min \quad & \sum_{f \in L} c_f x_f && && \text{(LP: Cover Small Cuts)} \\ \text{subject to:} \quad & \sum_{f \in L \cap \delta(S)} x_f \geq 1 && \forall\, S \in \mathcal{C} \\ & 0 \leq x_f \leq 1 && \forall\, f \in L. \end{aligned}$$

The first constant-factor approximation algorithm was presented by Bansal et al., [4], and the approximation ratio was improved from 16 to 10 by Nutov, [17], then from 10 to 5 by Bansal, [2]. The following result is due to Bansal, [2].

Proposition 3. *Given an instance of* Cover Small Cuts*, the WGMV primal-dual algorithm, [19], finds a feasible solution of cost $\leq 5\, LP_{opt}$ in polynomial time, where LP_{opt} denotes the optimal value of* (LP: Cover Small Cuts).

In section 3, we also discuss the 2-Cover Small Cuts problem. The inputs are the same as above, namely, $G = (V, E), u, L, c, \lambda$. A subset $F \subseteq L$ of links is said to *two-cover* a node-set S if $|F \cap \delta(S)| \geq 2$, that is, if there exist a pair of (distinct) links $e, e' \in F$ such that each of e and e' has exactly one end-node in S. The objective is to find a minimum-cost $F \subseteq L$ that two-covers each non-empty $S \subsetneq V$ with $u(\delta_E(S)) < \lambda$.

Our results

We present improved approximation algorithms for the (p, q)-FGC problem and the Cap-k-ECSS problem. In particular, we present the following results.

- A 7-approximation algorithm for the $(1, q)$-FGC problem.
- $O(1)$-approximate reductions between the $(2, q)$-FGC problem and the 2-Cover Small Cuts problem.
- An $O(\log k)$ approximation algorithm for the Cap-k-ECSS problem.
- Moreover, we present two auxiliary results in the appendices. The first one is motivated by the question: For what values of the parameters p and q can one formulate the (p, q)-FGC problem as an equivalent Cap-k-ECSS problem? We show that one can formulate a (p, q)-FGC problem as an equivalent Cap-k-ECSS problem if and only if $p = 1$ or $q = 1$. Our second auxiliary result gives a (randomized) $O(p \log n)$ approximation algorithm for the (p, q)-FGC problem, based on a result of Chekuri & Quanrud [9].

2 A 7-Approximation Algorithm for $(1, q)$-FGC

This section presents a 7-approximation algorithm for the $(1, q)$-FGC problem. For the convenience of the reader, the presentation in this section is independent of the rest of the paper.

Our starting point is the natural LP relaxation that follows from taking a capacitated network design view of the problem whereby we assign each unsafe edge $e \in \mathcal{U}$ has capacity $u_e = 1$, and each safe edge $e \in \mathcal{S}$ has capacity $u_e = (q + 1)$. The natural LP relaxation then seeks to minimize the total cost of edges subject to the constraint that $\forall\ \emptyset \subsetneq S \subsetneq V$, we have $\sum_{e \in \mathcal{S} \cap \delta(S)} (q + 1)x_e + \sum_{e \in \mathcal{U} \cap \delta(S)} x_e \geq (q + 1)$.

It is easy to see that any 0/1-valued solution to this LP is a valid solution to a given instance of $(1, q)$-FGC, and vice versa. However, it is also easy to show that this LP has an integrality gap of q by adapting the integrality gap example we saw earlier for the Cap-k-ECSS problem. We consider an instance consisting of a pair of nodes connected by q parallel unsafe edges of cost 0, and a single safe edge of cost 1. Now the optimal fractional solution has cost $1/q$ while the optimal integral solution has cost 1.

To get around this integrality gap, we strengthen the LP relaxation of $(1, q)$-FGC using the knapsack-cover inequalities to obtain the following stronger LP. Intuitively, the added knapsack-cover constraint ensures that if a safe edge is being used to cover a cut that is partly being covered by unsafe edges, say by k of them, then the capacity of the safe edge is reduced to $(q + 1) - k$.

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e & & \text{(KCLP:}(1,q)\text{-FGC)} \\
\text{s.t.} \quad & \sum_{e \in \mathcal{S} \cap \delta(S)} (q+1) x_e + \sum_{e \in \mathcal{U} \cap \delta(S)} x_e \geq (q+1) & & \forall\ \emptyset \subsetneq S \subsetneq V \\
& \sum_{e \in E \cap \delta(S) - A} u_e(A, S) x_e \geq D(A, S) & & \forall\ \emptyset \subsetneq S \subsetneq V, A \subseteq E \\
& 0 \leq x_e \leq 1 & & \forall\ e \in E,
\end{aligned}
$$

where $D(A, S) = \max\{0, (q + 1) - \sum\{u_e \mid e \in A \cap \delta(S)\}\}$, and $u_e(A, S) = \min\{u_e, D(A, S)\}$. We would like to solve the above LP using the ellipsoid method, but, unfortunately, we do not know a polynomial-time separation oracle for knapsack-cover inequalities. We will instead identify a subset of unsafe edges A, and a polynomial-time computable collection of cuts such that as long as the knapsack-cover inequalities hold for this collection, we will be able to show that the integrality gap of the fractional solution is at most 7. We can use this property to design a polynomial-time algorithm. In what follows, we assume w.l.o.g. that the optimal LP solution cost, say $\mathrm{LP}_{\mathrm{opt}}$, is known as it can be identified via binary search. We can thus replace the minimization objective with a feasibility constraint on our solution, namely, $\sum_{e \in E} c_e x_e \leq \mathrm{LP}_{\mathrm{opt}}$.

Lemma 4. *There is a polynomial-time algorithm that computes a solution x^* to (KCLP:$(1, q)$-FGC) of value at most LP_{opt} such that the solution satisfies the following two properties:*

(P1) $\sum_{e \in \mathcal{S} \cap \delta(S)} (q + 1)x^*_e + \sum_{e \in \mathcal{U} \cap \delta(S)} x^*_e \geq (q + 1)$.

(P2) *Let* $A = \{e \in \mathcal{U} \mid x^*_e \geq 2/7\}$*. For any nonempty* $S \subsetneq V$*, if* $\sum_{e\in\mathcal{S}\cap\delta(S)}(q+1)x^*_e + \sum_{e\in\mathcal{U}\cap\delta(S)} x^*_e \leq 2(q+1)$*, then*

$$\sum_{e\in E\cap\delta(S)-A} u_e(A,S)x_e \geq D(A,S).$$

Proof. It suffices to describe a polynomial-time separation oracle which identifies any violations of properties (P1) and (P2). Given a solution x^*, we first check that $\sum_{e\in E} c_e x^*_e \leq \mathrm{LP}_{\mathrm{opt}}$. If not, we return this as a violated constraint. Otherwise, let $\hat{G}(\hat{V},\hat{E})$ be the capacitated graph where $\hat{V} = V, \hat{E} = E$, and each edge $e \in \hat{E}$ is assigned a capacity of $u_e x^*_e$. We can now check that the capacity of a minimum-cut in G' is at least $(q+1)$ using a polynomial-time global minimum cut algorithm [18]. If not, we return a global minimum cut in $\hat{G}$ as a violated constraint. Otherwise, we know that (P1) is also satisfied, and we proceed to verify (P2) with respect to the set $A = \{e \in \mathcal{U} \mid x^*_e \geq 2/7\}$.

By Karger's result [15], we know that there are at most $O(n^4)$ cuts of capacity at most $2(q+1)$ (i.e., at most twice the capacity of a minimum-cut), and, moreover, we can enumerate all such cuts of $\hat{G}$ in polynomial time [16]. By iterating over each of the $O(n^4)$ cuts, we can now verify in polynomial-time that the knapsack cover inequalities are satisfied w.r.t. the set A. If not, we have found a violated constraint.

Since the ellipsoid algorithm terminates after $n^{O(1)}$ iterations of feasibility verification [13], this gives us a polynomial-time algorithm that computes a solution x^* to (KCLP:$(1,q)$-FGC) of value at most $\mathrm{LP}_{\mathrm{opt}}$ such that the solution satisfies properties (P1) and (P2). □

The Rounding Algorithm: Given a solution x^* of cost at most $\mathrm{LP}_{\mathrm{opt}}$ that satisfies (P1) and (P2), Algorithm 1, presented below, rounds it to an integral solution of cost at most $7\mathrm{LP}_{\mathrm{opt}}$. We describe here the main idea of our rounding scheme.

We say a non-trivial cut $\delta(S)$ is a *small cut* if $\sum_{e\in\mathcal{U}_1\cap\delta(S)} u_e + \sum_{e\in\mathcal{U}_2\cap\delta(S)} \frac{7}{2}u_e x^*_e < (q+1)$ where $\mathcal{U}_1 = \{e \in \mathcal{U} : x^*_e \geq 2/7\}$, and $\mathcal{U}_2 = \mathcal{U} \setminus \mathcal{U}_1$. To handle the presence of small cuts, we will first show that the solution x^* restricted to safe edges and scaled up by a factor of 7/5 constitutes a feasible solution to the Cover Small Cuts instance defined by small cuts (see Lemma 5). We can thus pick a set $\mathcal{S}_1$ of safe edges by applying the 5-approximation algorithm of [2, 4] to this instance of the Cover Small Cuts problem, getting a solution whose cost is at most 7 times as large as the cost of x^* restricted to safe edges. After this step, we can contract connected components formed by edges in $\mathcal{S}_1$, and get a new instance that does not have any small cuts. Since there are no small cuts, we can get a feasible solution for the f-connectivity problem where $f(S) = q+1$ for every non-trivial cut $\delta(S)$ by taking all edges in $\mathcal{U}_1$ and scaling up x^* restricted to $\mathcal{U}_2$ by a factor of 7/2. Since f is weakly supermodular, we can apply Jain's iterative rounding scheme [14] to solve this f-connectivity problem and recover a 2-approximate solution, giving us an integral solution whose cost is at most 7 times as large as the cost of x^* restricted to unsafe edges (see Lemma 6). Thus, we get an integral solution whose total cost of safe and unsafe edges is bounded by at most 7 times the LP cost.

The next two lemmas formalize the key properties of the solution x^* that are used in the rounding scheme above, allowing us to show that it indeed returns a feasible integral solution of cost at most 7 $c(x^*)$.

Lemma 5. *In step 2 of Algorithm 1, a feasible fractional solution to the CoverSmallCuts instance is given by* $\hat{x}_e = \min\{1, \frac{7}{5}x^*_e\}$ *for* $e \in \mathcal{S}$*.*

Proof. Fix any small cut $\delta(S)$. We will establish the lemma by considering two cases below.

Consider first the case that $\sum_{e\in\mathcal{S}\cap\delta(S)}(q+1)x^*_e + \sum_{e\in\mathcal{U}\cap\delta(S)} x^*_e > 2(q+1)$. Since $\delta(S)$ is a small cut, we have

$$\sum_{e\in\mathcal{U}_1\cap\delta(S)} x^*_e + \sum_{e\in\mathcal{U}_2\cap\delta(S)} x^*_e \leq \sum_{e\in\mathcal{U}_1\cap\delta(S)} u_e + \sum_{e\in\mathcal{U}_2\cap\delta(S)} \frac{7}{2}u_e x^*_e < (q+1).$$

Since $\sum_{e\in\mathcal{S}\cap\delta(S)}(q+1)x^*_e + \sum_{e\in\mathcal{U}\cap\delta(S)} x^*_e > 2(q+1)$, it follows that $\sum_{e\in\mathcal{S}\cap\delta(S)}(q+1)x^*_e \geq (q+1)$, and hence $\sum_{e\in\mathcal{S}\cap\delta(S)}(q+1)x^*_e \geq 1$. Thus $\sum_{e\in\mathcal{S}\cap\delta(S)} \hat{x}_e \geq 1$.

Now suppose that $\sum_{e\in\mathcal{S}\cap\delta(S)}(q+1)x^*_e + \sum_{e\in\mathcal{U}\cap\delta(S)} x^*_e \leq 2(q+1)$. Then by (P2) we know that the cut $\delta(S)$ satisfies knapsack-cover constraint w.r.t. set $A = \mathcal{U}_1$:

$$\sum_{e\in\mathcal{S}\cap\delta(S)} u_e(A,S)x_e + \sum_{e\in\mathcal{U}_2\cap\delta(S)} u_e(A,S)x_e \geq D(A,S),$$

Algorithm 1: 7-approximate solution to $(1, q)$-FGC

Require: Graph $G = (V, E)$ where E is partitioned into $\mathcal{S} \bigcup \mathcal{U}$, with edge costs $\{c_e\}_{e \in E}$. A solution x^* to (KCLP:$(1, q)$-FGC) as promised by Lemma 4.

1: $\mathcal{U}_1 \leftarrow \{e \in \mathcal{U} : x^*_e \geq 2/7\}$, and $\mathcal{U}_2 \leftarrow \mathcal{U} - \mathcal{U}_1$.

2: (a) Apply the approximation algorithm for Cover Small Cuts on the instance with $G' = (V, E' = \mathcal{U})$, where each edge of $\mathcal{U}_1$ is given unit capacity, each edge $e \in \mathcal{U}_2$ is given capacity $7/2\, x^*_e$, and with link set $\mathcal{S}$. Define the threshold λ (for Small Cuts) to be $(q + 1)$. Thus, we have:

$$\sum_{e \in \mathcal{U}_1 \cap \delta(S)} u_e + \sum_{e \in \mathcal{U}_2 \cap \delta(S)} \frac{7}{2} u_e x^*_e < (q + 1) \qquad \text{(definition of } \textit{small cuts}\text{)}$$

(b) Let the output of the call in step (a) be denoted $\mathcal{S}_1$.

3: Construct a graph $G'' = (V, \mathcal{U}_2)$ where each edge $e \in \mathcal{U}_2$ has cost c_e. Define the requirement of a non-trivial cut $\delta(S)$ to be

$$f(S) = \max\{0, \;\; (q + 1) - ((q + 1)|\delta(S) \cap \mathcal{S}_1| + |\delta(S) \cap \mathcal{U}_1|)\}.$$

This function f is weakly supermodular, so a 2-approximate solution for this instance of f-connectivity can be computed using Jain's iterative rounding method [14].

4: Return the union of the set of unsafe edges picked by the previous step (via the iterative rounding algorithm) and $\mathcal{S}_1 \cup \mathcal{U}_1$.

---

where $D(A, S) = \max\{0, (q + 1) - \sum\{u_e \mid e \in A \cap \delta(S)\}\} = \max\{0, (q + 1) - |\mathcal{U}_1 \cap \delta(S)|\}$, $u_e(A, S) = \min\{u_e, D(A, S)\}$.

Moreover, since $\delta(S)$ is a small cut, we know that $|\mathcal{U}_1 \cap \delta(S)| + \sum_{e \in \mathcal{U}_2 \cap \delta(S)} \frac{7}{2} x^*_e < (q + 1)$, and hence $\sum_{e \in \mathcal{U}_2 \cap \delta(S)} \frac{7}{2} u_e(A, S) x^*_e < D(A, S)$. This implies that $\sum_{e \in \mathcal{U}_2 \cap \delta(S)} u_e(A, S) x^*_e < \frac{2}{7} D(A, S)$, and so, by the knapsack-cover inequality, $\sum_{e \in \mathcal{S} \cap \delta(S)} u_e(A, S) x^*_e \geq \frac{5}{7} D(A, S)$. By definition, $u_e(A, S) \leq D(A, S)$, hence, we have $\sum_{e \in \mathcal{S} \cap \delta(S)} \frac{7}{5} D(A, S) x^*_e \geq D(A, S)$. Therefore, $\sum_{e \in \mathcal{S} \cap \delta(S)} \hat{x}_e \geq 1$, completing the proof. □

Lemma 6. *In step 3 of Algorithm 1, a feasible fractional solution to the f-connectivity problem is given by $x'_e = \frac{7}{2} x^*_e$ for $e \in \mathcal{U}_2$.*

Proof. By way of contradiction, suppose that the claim does not hold. Then for some non-trivial cut $\delta(S)$, we must have

$$\sum_{e \in (\mathcal{S}_1 \cup \mathcal{U}_1) \cap \delta(S)} u_e + \sum_{e \in \mathcal{U}_2 \cap \delta(S)} \frac{7}{2} u_e x^*_e < (q + 1).$$

This implies that the cut $\delta(S)$ is a *small cut* in step 2 of Algorithm 1. Hence, step 2 ensures (via CoverSmallCuts) that $|\mathcal{S}_1 \cap \delta(S)| \geq 1$ and so $\sum_{e \in (\mathcal{S}_1 \cup \mathcal{U}_1) \cap \delta(S)} u_e \geq (q + 1)$. This is a contradiction. □

The output of Algorithm 1 is feasible for the $(1, q)$-FGC problem due to the definition of the f-connectivity problem in step 3 of the algorithm. The cost of the edges in $\mathcal{U}_1$ is $\leq 7/2 \sum_{e \in \mathcal{U}_1} x^*_e c_e$ since $x^*_e \geq 2/7$ for each edge e in $\mathcal{U}_1$. Additionally, the cost of the edges in $\mathcal{S}_1$ is $\leq 7 \sum_{e \in \mathcal{S}_1} x^*_e c_e$ by Lemma 5 and by Proposition 3. Lastly, the cost of the edges returned by Jain's iterative rounding algorithm in step 3 of Algorithm 1 is at most $\sum_{e \in \mathcal{U}_2} (2)(7/2) c_e x^*_e = 7 \sum_{e \in \mathcal{U}_2} c_e x^*_e$, by Lemma 6. Putting it all together, the cost of the solution returned by Algorithm 1 is at most $7\ c(x^*)$.

Theorem 7. *Let* OPT *be the optimal solution value for a given instance of* $(1, q)$-FGC. *Then there is a polynomial-time algorithm that computes a solution x^* to* (KCLP:$(1, q)$-FGC) *of value at most* OPT *(possibly satisfying only a subset of the constraints) and rounds it to obtain a feasible integer solution of cost at most* 7 OPT.

3 $O(1)$-Approximate Reductions between $(2,q)$-FGC and 2-Cover Small Cuts

We provide reductions between the two problems $(2,q)$-FGC and 2-Cover Small Cuts in both directions. Each of these reductions preserves the approximation ratio up to a constant factor.

Recall that in the 2-Cover Small Cuts problem, we are given a graph $\hat{G} = (\hat{V}, \hat{E})$ with edge capacities u, a number λ, as well as a set of links $L \subseteq \binom{\hat{V}}{2}$ with link costs c; the goal is to find a cheapest set of links $J \subseteq L$ that two-covers the family of small cuts, namely, $\{S \subsetneq \hat{V}, S \neq \emptyset \mid u(\delta(S)) < \lambda\}$.

Reducing $(2,q)$-FGC to 2-Cover Small Cuts with approximation ratio $O(1)$

Suppose we have an LP relative α-approximation algorithm for 2-Cover Small Cuts. (In other words, suppose we have access to a polynomial-time algorithm that rounds a given fractional feasible solution of the LP relaxation of 2-Cover Small Cuts to a feasible (integral) solution of cost $\leq \alpha\, \mathrm{LP_{opt}}$, where $\mathrm{LP_{opt}}$ denotes the optimal value of the LP relaxation.)

Let $G = (V, E = \mathcal{S} \cup \mathcal{U})$ be an instance of $(2,q)$-FGC with edge costs $c \in \mathbb{Q}^E_{\geq 0}$. We use the following linear programming relaxation. Informally speaking, the first type of constraints (see (1) below) correspond to a $(1, q+1)$-FGC problem, and the second type of constraints (see (2) below) say that for each safe edge f, the edges in $E - f$ should satisfy the requirements of $(1,q)$-FGC (i.e., each nontrivial cut has one safe edge or $q+1$ edges).

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e && \text{(LP:}(2,q)\text{-FGC)}\\
\text{s.t.} \quad & \sum_{e \in \mathcal{S} \cap \delta(S)} (q+2) x_e + \sum_{e \in \mathcal{U} \cap \delta(S)} x_e \geq q+2 && \forall\ \emptyset \subsetneq S \subsetneq V \quad (1)\\
& \sum_{e \in (\mathcal{S}-f) \cap \delta(S)} (q+1) x_e + \sum_{e \in \mathcal{U} \cap \delta(S)} x_e \geq q+1 && \forall\ \emptyset \subsetneq S \subsetneq V,\ \forall\ f \in \mathcal{S} \quad (2)\\
& 0 \leq x \leq 1
\end{aligned}$$

Each of the inequalities (1) and each of the inequalities (2) (for each $f \in \mathcal{S}$, $\forall S \subsetneq V, S \neq \emptyset$) can be strengthened using the knapsack-cover inequalities.

We follow the method of section 2 for rounding this strengthened LP. (Recall that no polynomial-time separation algorithm is known for these knapsack-cover inequalities, hence, we have to work around this gap.) We assume w.l.o.g. that $\mathrm{LP_{opt}}$, the optimal value of the strengthened LP, is known, since it can be identified via binary search. Thus, we replace the minimization objective with a feasibility constraint, namely, $\sum_{e \in E} c_e x_e \leq \mathrm{LP_{opt}}$. As in section 2, we will design a polynomial-time algorithm for rounding the strengthened LP by picking some sets of unsafe edges, and a polynomial-time computable collection of cuts $\mathcal{C}$. We will show that our rounding algorithm succeeds as long as the knapsack-cover inequalities hold for each set $S \in \mathcal{C}$ and an appropriate set of unsafe edges A.

Lemma 8. *There is a polynomial-time algorithm that computes a solution x^* to (LP:$(2,q)$-FGC) such that $\sum_e c_e x^*_e \leq LP_{opt}$ and, moreover, x^* satisfies the following properties:*

(P3) $\sum_{e \in \mathcal{S} \cap \delta(S)} (q+2) x^*_e + \sum_{e \in \mathcal{U} \cap \delta(S)} x^*_e \geq (q+2) \qquad \forall\, \emptyset \neq S \subsetneq V$.

(P4) Let $A^{(1)} = \{e \in \mathcal{U} \mid x^*_e \geq 2/7\}$. *For any nonempty* $S \subsetneq V$, *if* $\sum_{e \in \mathcal{S} \cap \delta(S)} (q+2) x^*_e + \sum_{e \in \mathcal{U} \cap \delta(S)} x^*_e \leq 2(q+2)$, *then*

$$\sum_{e \in E \cap \delta(S) - A^{(1)}} u_e(A^{(1)}, S) x_e \geq D(A^{(1)}, S).$$

(P5) For each safe edge f, $\sum_{e \in (\mathcal{S}-f) \cap \delta(S)} (q+1) x^*_e + \sum_{e \in \mathcal{U} \cap \delta(S)} x^*_e \geq (q+1) \qquad \forall\, \emptyset \neq S \subsetneq V$.

(P6) *Let* $A^{(2)} = \{e \in \mathcal{U} \mid x^*_e \geq 1/2\}$. *For each safe edge* f *and for any nonempty* $S \subsetneq V$, *if* $\sum_{e \in (\mathcal{S}-f)\cap\delta(S)} (q+1)x^*_e + \sum_{e\in\mathcal{U}\cap\delta(S)} x^*_e \leq 2(q+1)$, *then*

$$\sum_{e \in E\cap\delta(S)-A^{(2)}} u_e(A^{(2)}, S)x_e \geq D(A^{(2)}, S).$$

Proof. We describe a polynomial-time separation algorithm that identifies any violation of properties (P3)–(P6). Given a vector x^*, we first check that $\sum_{e\in E} c_e x^*_e \leq \mathrm{LP}_{\mathrm{opt}}$. If not, we return this as a violated constraint. Otherwise, let $\hat{G}(\hat{V}, \hat{E})$ be the capacitated graph where $\hat{V} = V, \hat{E} = E$, and each edge $e \in \hat{E}$ is assigned a capacity of $u_e x^*_e$.

By following the method of the proof of Lemma 4, we can verify that either (P3) and (P4) hold or we can return a violated constraint.

Finally, we consider (P5) and (P6). Consider any safe edge f, and the capacitated graph $\hat{G} - f$. First, we check that the minimum-cut capacity in $\hat{G} - f$ is $\geq (q + 1)$ using a polynomial-time minimum-cut algorithm [18]. If not, we return a minimum cut in $\hat{G} - f$ as a violated constraint. Otherwise, we know that (P5) is satisfied, and we proceed to verify (P6) with respect to the set $A^{(2)}$.

By Karger's results, $\hat{G} - f$ has at most $O(n^4)$ cuts of capacity $\leq 2(q + 1)$ (i.e. at most twice the minimum-cut capacity); moreover, we can enumerate all such cuts in polynomial time [16]. By iterating over each of the $O(n^4)$ cuts $\delta(S)$, we can verify in polynomial-time whether the knapsack-cover inequality for the set $S \subsetneq V$ and $A^{(2)} \subseteq \mathcal{U}$ is satisfied. If not, we have found a violated constraint.

Since the ellipsoid algorithm terminates after $n^{O(1)}$ iterations of feasibility verification [13], this gives us a polynomial-time algorithm that computes a solution x^* to (LP:$(2, q)$-FGC) of value $\leq \mathrm{LP}_{\mathrm{opt}}$ such that x^* satisfies properties (P3)–(P6). $\square$

Our algorithm for rounding x^* has two parts. The first part applies the algorithm of section 2 and finds a set of edges $\mathcal{S}_0 \cup \mathcal{U}_0$ that is feasible for the $(1, q + 1)$-FGC instance. The second part uses the (assumed) α-approximation algorithm for 2-Cover Small Cuts, as well as Jain's iterative rounding method, and finds a set of edges $\mathcal{S}_1 \cup \mathcal{S}_{\mathrm{ALG}} \cup J$. Below, we show that the union of the two sets of edges is feasible for $(2, q)$-FGC and it has cost $\leq (4\alpha + 11)\, c(x^*)$.

Next, we present the details of our algorithm and its analysis.

The first part simply applies the algorithm of section 2 to round x^* to an edge set $\mathcal{S}_0 \cup \mathcal{U}_0$ of cost $\leq 7\, c(x^*)$ that is feasible for the $(1, q+1)$-FGC problem (described by the inequalities (1)), where $\mathcal{S}_0$ is a set of safe edges and $\mathcal{U}_0$ is a set of unsafe edges. (We mention that the fact that x^* satisfies the inequalities (2) is not relevant for this part.)

In the second part of the algorithm, we start by defining $\mathcal{S}_1 = \{e \in \mathcal{S} : x^*_e \geq 1/4\}$ and $\mathcal{U}_1 = \{e \in \mathcal{U} : x^*_e \geq 1/2\}$. Let $\mathcal{S}_2 = \mathcal{S} - \mathcal{S}_1$ and let $\mathcal{U}_2 = \mathcal{U} - \mathcal{U}_1$.

Define $\mathcal{C}$ to be the family of all nontrivial cuts $\delta(S)$ such that $|\mathcal{U}_1 \cap \delta(S)| + \sum_{e\in\mathcal{U}_2\cap\delta(S)} 2x^*_e < (q + 1)$.

The analysis of the second part of our algorithm hinges on the next lemma. It shows that a (small) constant multiple of the fractional vector $x^*_{\mathcal{S}}$ is feasible for the LP relaxation of the 2-Cover Small Cuts instance defined by $\mathcal{C}$.

Lemma 9. *Let* x^* *satisfy properties (P3)–(P6) of Lemma 8. A feasible fractional solution to the* 2-Cover Small Cuts *instance defined by* $\mathcal{C}$ *is given by the vector* $(\mathbb{1}_{\mathcal{S}_1}, 4x^*_{\mathcal{S}_2})$; *that is, for any* $\delta(S) \in \mathcal{C}$, *we have* $|\mathcal{S}_1 \cap \delta(S)| + \sum_{e\in\mathcal{S}_2\cap\delta(S)} 4x^*_e \;\geq\; 2$.

Proof. We start with a key claim.

Claim 10. *Consider any safe edge* f *and any nontrivial cut* $\delta(S)$ *of* $\mathcal{C}$. *Then we have*

$$\sum_{e\in(\mathcal{S}-f)\cap\delta(S)} x^*_e \geq 1/2.$$

Proof. We prove this claim by examining two cases, based on property (P6). First, suppose $\sum_{e\in(\mathcal{S}-f)\cap\delta(S)}(q+1)x^*_e + \sum_{e\in\mathcal{U}\cap\delta(S)} x^*_e > 2(q+1)$. Then (since $\delta(S)$ is in $\mathcal{C}$) we have

$$\sum_{e\in\mathcal{U}_1\cap\delta(S)} x^*_e + \sum_{e\in\mathcal{U}_2\cap\delta(S)} x^*_e \leq \sum_{e\in\mathcal{U}_1\cap\delta(S)} u_e + \sum_{e\in\mathcal{U}_2\cap\delta(S)} 2u_e x^*_e < (q+1).$$

Hence, $\sum_{e\in(\mathcal{S}-f)\cap\delta(S)}(q+1)x^*_e \geq (q+1)$, and it follows that $\sum_{e\in(\mathcal{S}-f)\cap\delta(S)} x^*_e \geq 1$. Thus, the claim holds in the first case.

In the second case, we have $\sum_{e\in(\mathcal{S}-f)\cap\delta(S)}(q+1)x^*_e + \sum_{e\in\mathcal{U}\cap\delta(S)} x^*_e \leq 2(q+1)$. We will use the fact that x^* satisfies property (P6). Let $A = \mathcal{U}_1$. The knapsack-cover constraint for the cut $\delta(S)$ and A is

$$\sum_{e\in(\mathcal{S}-f)\cap\delta(S)} u_e(A,S)x_e + \sum_{e\in\mathcal{U}_2\cap\delta(S)} u_e(A,S)x_e \geq D(A,S),$$

where $D(A,S) = \max\{0,\, q+1-|\mathcal{U}_1\cap\delta(S)|\}$, $u_e(A,S) = \min\{u_e,\, D(A,S)\}$, each edge $e\in\mathcal{U}_2$ has $u_e = 1$, and each safe edge e has $u_e = q+1$. Since $\delta(S)$ is in $\mathcal{C}$, we have the inequality $|\mathcal{U}_1\cap\delta(S)| + \sum_{e\in\mathcal{U}_2\cap\delta(S)} 2x^*_e < q+1$, and this inequality can be rewritten in the form $\sum_{e\in\mathcal{U}_2\cap\delta(S)} 2u_e(A,S)x^*_e < D(A,S)$. This implies that $\sum_{e\in\mathcal{U}_2\cap\delta(S)} u_e(A,S)x^*_e < \frac{1}{2}D(A,S)$, and so, by the knapsack-cover inequality, $\sum_{e\in(\mathcal{S}-f)\cap\delta(S)} u_e(A,S)x^*_e \geq \frac{1}{2}D(A,S)$. Since $u_e(A,S) \leq D(A,S)$, we have $\sum_{e\in(\mathcal{S}-f)\cap\delta(S)} D(A,S)x^*_e \geq \frac{1}{2}D(A,S)$, hence, $\sum_{e\in(\mathcal{S}-f)\cap\delta(S)} x^*_e \geq \frac{1}{2}$. The claim follows. □

Let us fix any nontrivial cut $\delta(S)$ in $\mathcal{C}$. We will prove the lemma by examining a few cases. First, suppose $|\mathcal{S}_1\cap\delta(S)| \geq 2$. Then the inequality is trivially true. Second, suppose $|\mathcal{S}_1\cap\delta(S)| = 1$. Then, Claim 10 above implies that $\sum_{e\in\mathcal{S}_2\cap\delta(S)} 4x^*_e \geq 2$. Lastly, suppose $\mathcal{S}_1\cap\delta(S) = \emptyset$. Then again Claim 10 above implies that $\sum_{e\in\mathcal{S}_2\cap\delta(S)} 4x^*_e \geq 2$. This proves the lemma. □

By Lemma 9, we can use the α-approximation algorithm for 2-Cover Small Cuts to find a set of safe edges, call this set $\mathcal{S}_{\text{ALG}}$, that two-covers all the cuts in $\mathcal{C}$, incurring a cost of $\leq 4\alpha\, c(x^*_{\mathcal{S}})$.

Now, for every nontrivial cut $\delta(S)$, we either have $|\delta(S)\cap\mathcal{S}_{\text{ALG}}| \geq 2$, or we have $|\delta(S)\cap\mathcal{U}_1| + \sum_{e\in\mathcal{U}_2\cap\delta(S)} 2x^*_e \geq (q+1)$.

Next, we formulate an f-connectivity problem such that a feasible solution to this problem will cover the nontrivial cuts $\delta(S)$ with $|\delta(S)\cap\mathcal{S}_{\text{ALG}}| < 2$.

For a nontrivial cut $\delta(S)$ define,

$$f(S) = \begin{cases} q+1, & \text{if } \emptyset \neq S \subsetneq V \text{ and } |\delta(S)\cap\mathcal{S}_{\text{ALG}}| \leq 1 \\ 0, & \text{otherwise.} \end{cases}$$

Observe that $(1_{\mathcal{U}_1}, 2x^*_{\mathcal{U}_2})$ is a fractional feasible solution to the f-connectivity problem. By Proposition 2, Jain's iterative rounding algorithm finds a feasible solution $J \subseteq \mathcal{U}$ to this f-connectivity problem of cost $\leq 4c(x^*_{\mathcal{U}})$.

Finally, we claim that the union of the edge sets found by the two parts of the algorithm, namely, $E_{\text{ALG}} = (\mathcal{S}_0\cup\mathcal{U}_0)\bigcup(\mathcal{S}_1\cup\mathcal{S}_{\text{ALG}}\cup\mathcal{U}_1\cup J)$ is a feasible solution of the $(2,q)$-FGC instance, that is, every nontrivial cut either has two safe edges (of E_{ALG}) or has $q+2$ edges (of E_{ALG}). Consider any nontrivial cut $\delta(S)$ with at most one safe edge (of E_{ALG}). If $\delta(S)\cap E_{\text{ALG}}$ has one safe edge, then $|\delta(S)\cap J| \geq q+1$ (since J is feasible for the f-connectivity problem). If $\delta(S)\cap E_{\text{ALG}}$ has no safe edges, then $|\delta(S)\cap\mathcal{U}_0| \geq q+2$ (since $\mathcal{S}_0\cup\mathcal{U}_0$ is feasible for the $(1,q+1)$-FGC problem).

Clearly, the cost of E_{ALG} is $\leq 7\,c(x^*) + 4\alpha\, c(x^*_{\mathcal{S}}) + 4\,c(x^*_{\mathcal{U}}) \leq (4\alpha+11)\,c(x^*)$.

Theorem 11. *Suppose a polynomial-time LP relative α-approximation algorithm for* 2-Cover Small Cuts *is available.*

Then the above algorithm for $(2,q)$-FGC *runs in polynomial time and returns a feasible (integral) solution of cost* $\leq (4\alpha+11)\,\text{OPT}$*, where* OPT *denotes the optimal value of the* $(2,q)$-FGC *instance.*

Reducing 2-Cover Small Cuts to $(2,q)$-FGC with approximation ratio $O(1)$

In this subsection, we show that a β-approximation algorithm for $(2,q)$-FGC implies an $(\beta+2)$-approximation algorithm for unit-capacity 2-Cover Small Cuts.

Let $\hat{G}=(\hat{V},\hat{E})$, u, L, c, λ be an instance of the unit-capacity 2-Cover Small Cuts problem; thus, each edge $e\in\hat{E}$ has $u_e=1$. The goal is to find a cheapest set of links that two-covers all the nontrivial cuts $\delta(S)$ such that $|\delta(S)\cap\hat{E}|<\lambda$. Let L^* denote an optimal solution, and let $\textsc{opt}=c(L^*)$ denote its cost.

We map the above instance of 2-Cover Small Cuts to an instance of $(2,q)$-FGC (in general, the two instances are not equivalent since the two problems are different). The graph is $G=(V,E=\mathcal{S}\cup\mathcal{U})$ with $V=\hat{V}$, $\mathcal{U}=\hat{E}$, and $\mathcal{S}=L$. Each unsafe edge has cost zero, and each safe edge has the cost of the corresponding link (in the 2-Cover Small Cuts instance). We fix the parameter q to be $\lambda-2$. Let us use the same notation for a set of links (of the 2-Cover Small Cuts instance) and the corresponding set of safe edges (of the $(2,q)$-FGC instance).

A feasible solution to this $(2,q)$-FGC instance picks a subset J of the safe edges such that J two-covers all nontrivial cuts $\delta(S)$ with $|\delta(S)\cap\mathcal{U}|\leq\lambda-2$, and J covers all nontrivial cuts $\delta(S)$ with $|\delta(S)\cap\mathcal{U}|=\lambda-1$.

Observe that $L^*\cup\mathcal{U}$ is a feasible solution (of the $(2,q)$-FGC instance), and it has cost $\textsc{opt}$. Hence, our β-approximation algorithm for $(2,q)$-FGC finds a feasible solution of cost $\leq\beta\,\textsc{opt}$. Let $\mathcal{S}_{\text{ALG}}$ denote the set of safe edges picked by our β-approximation algorithm.

In order to obtain a feasible solution for the 2-Cover Small Cuts instance, we need to augment $\mathcal{S}_{\text{ALG}}$ by a set of safe edges J such that J covers all cuts $\delta(S)$ such that $|\delta(S)\cap\mathcal{U}|=\lambda-1$ and $|\delta(S)\cap\mathcal{S}_{\text{ALG}}|=1$. We achieve this subgoal via the following f-connectivity problem. For a cut $\delta(S)$ define,

$$f(S)=\begin{cases}\lambda, & \text{if } \emptyset\neq S\subsetneq V \text{ and } |\delta(S)\cap\mathcal{S}_{\text{ALG}}|\leq 1\\ 0, & \text{otherwise.}\end{cases}$$

The graph for this f-connectivity problem is the subgraph $G'=G-\mathcal{S}_{\text{ALG}}=(V,\mathcal{U}\cup(\mathcal{S}-\mathcal{S}_{\text{ALG}}))$, and the edge costs are as in G (that is, each unsafe edge has cost zero and each safe edge of G' has the cost of the corresponding link of the 2-Cover Small Cuts instance).

Observe that $J^*=\mathcal{U}\cup(L^*-\mathcal{S}_{\text{ALG}})$ is a feasible solution to this f-connectivity problem. We verify this claim by considering any nontrivial cut $\delta(S)$ and examining three cases: (i) If $|\delta(S)\cap\mathcal{U}|\geq\lambda$, then clearly $|\delta(S)\cap J^*|\geq f(S)$ (thus the requirement is satisfied); (ii) If $|\delta(S)\cap\mathcal{U}|\leq\lambda-2$, then $|\delta(S)\cap\mathcal{S}_{\text{ALG}}|\geq 2$, hence, $f(S)=0$ (thus there is no requirement); (iii) If $|\delta(S)\cap\mathcal{U}|=\lambda-1$, then either $|\delta(S)\cap\mathcal{S}_{\text{ALG}}|\geq 2$ and $f(S)=0$ or $|\delta(S)\cap\mathcal{S}_{\text{ALG}}|=1$ and $f(S)=\lambda$ and $|\delta(S)\cap J^*|\geq f(S)$ (because $|\delta(S)\cap L^*|\geq 2>|\delta(S)\cap\mathcal{S}_{\text{ALG}}|$).

By Proposition 2, Jain's iterative rounding algorithm finds a feasible solution $J'\subseteq\mathcal{S}-\mathcal{S}_{\text{ALG}}$ to this f-connectivity problem of cost $\leq 2\textsc{opt}$.

Observe that $\mathcal{S}_{\text{ALG}}\cup J'$ is a feasible solution to the 2-Cover Small Cuts instance because any nontrivial cut $\delta(S)$ with $|\delta(S)\cap\mathcal{U}|=|\delta(S)\cap\hat{E}|\leq\lambda-2$ is two-covered by $\mathcal{S}_{\text{ALG}}$, and any other nontrivial cut $\delta(S)$ with $|\delta(S)\cap\mathcal{U}|=|\delta(S)\cap\hat{E}|<\lambda$ is covered (once) by $\mathcal{S}_{\text{ALG}}$ and is covered (once) by $J'-\mathcal{S}_{\text{ALG}}$.

Thus we obtain a feasible solution to the 2-Cover Small Cuts instance with cost $\leq(\beta+2)\,\textsc{opt}$.

Theorem 12. *Suppose a polynomial-time β-approximation algorithm for* $(2,q)$-FGC *is available. Then the above algorithm for* 2-Cover Small Cuts *runs in polynomial time and returns a feasible solution of cost $\leq(\beta+2)\,\textsc{opt}$, where $\textsc{opt}$ denotes the optimal value of the* 2-Cover Small Cuts *instance.*

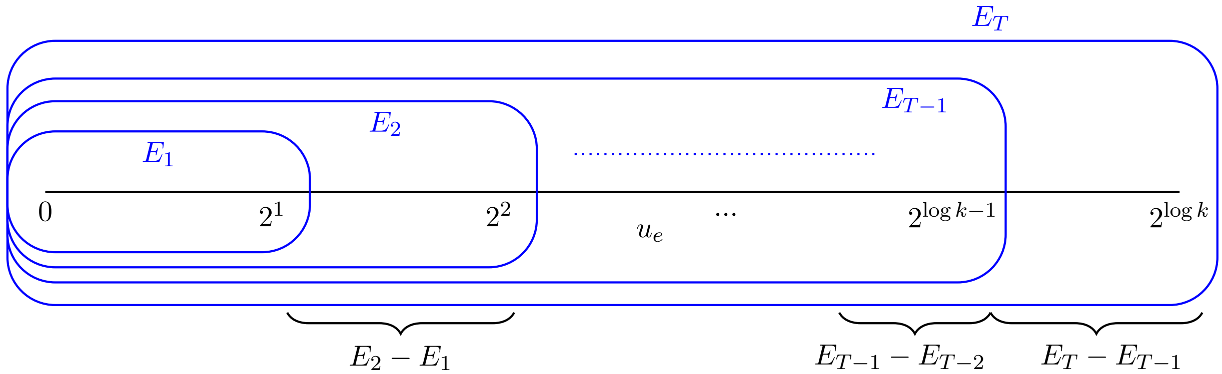
Remark. *We could not extend these reductions to* (p,q)-FGC *problems for $p>2$; one difficulty is that we could not design $O(1)$ approximation algorithms for the appropriate f-connectivity problems. In particular, we are not able to show similar reductions between $(3,q)$-FGC and* 3-Cover Small Cuts.

4 An $O(\log k)$-Approximation Algorithm for Cap-k-ECSS

In this section, we present an $O(\log k)$-approximation algorithm for Cap-k-ECSS that runs in polynomial-time assuming $k\leq|V(G)|=n$. Note that when $k>n$, then the previously known approximation algorithm of [7] for Cap-k-ECSS achieves an approximation ratio of $O(\log n)\leq O(\log k)$.

Let x^* be an optimal solution to (KCLP: CapkECSS), namely, the natural LP relaxation strengthened with knapsack-cover inequalities. We describe below our rounding algorithm. For the presentation of the rounding algorithm, it will be convenient to assume that x^* satisfies all constraints in (KCLP: CapkECSS) even though we do not know a polynomial-time separation oracle for achieving this. However, as we will show in Lemma 13, we can compute in poly-time a solution of optimal cost that satisfies all knapsack-cover inequalities that are needed for a successful execution of the rounding algorithm.

Throughout the execution of the rounding algorithm, we will maintain a set of edges E_{cur} acting as our current solution. We begin with $E_{cur} = \{e \in E : x^*_e \geq 1/2\}$. Define $T = \lceil \log k \rceil$ and for $j = 1, 2, \ldots, T$, define $E_j = \{e \in E : x^*_e < 1/2 \text{ and } u_e \leq 2^j\}$; thus, $E_T - E_{T-1}, E_{T-1} - E_{T-2}, \ldots, E_3 - E_2, E_2 - E_1, E_1$ forms a partition of the edges in $E - E_{cur}$ into T buckets based on the capacities; let us call the set $E_{T-i+1} - E_{T-i}$ the i-th bucket (and E_1 is the T-th bucket). See Figure 4 for an illustration.



Our algorithm will have T iterations and each iteration (except for the first and the last) will have two phases. During iteration i, we will be rounding some of the edges in the i-th bucket, i.e., some of the edges in the set $E_{T-i+1} - E_{T-i}$. Note that an edge e in the i-th bucket has capacity $2^{T-i} < u_e \leq 2^{T-i+1}$.

For every non-trivial cut S, we will maintain the following invariants for all iterations $i \in [2..(T-1)]$ (i.e. except the first and the last iteration):

1. At the beginning of iteration i, $E_{cur} \cap E_{T-i+1} = \emptyset$ and
$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i+1} \cap \delta(S)} 2\ u_e x^*_e \geq k.$$
We note that iteration 1 ensures that this invariant holds at the start of iteration 2.

2. At the end of the first phase of iteration i,
$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i} \cap \delta(S)} 2\ u_e x^*_e \geq k - 2^{T-i+1} - 2^{T-i}.$$

3. At the end of iteration i, which is also the end of phase 2 of iteration i, $E_{cur} \cap E_{T-i} = \emptyset$, and
$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i} \cap \delta(S)} 2\ u_e x^*_e \geq k.$$

Observe that invariant 3 for iteration i is the same as invariant 1 for iteration $i+1$.

Next, we present pseudo-code for the rounding algorithm, followed by explanation and analysis of the main steps.

Algorithm 2: $O(\log(k))$-approximate solution to Cap-k-ECSS

Require: Graph $G = (V, E)$ with edge capacities $\{u_e\}_{e\in E}$ and costs $\{c_e\}_{e\in E}$. A solution x^* to (KCLP: CapkECSS), set $E_{cur} = \{e \in E : x^*_e \geq 1/2\}$, and sets $E_j \subseteq E, j = 1, ..., \lceil \log k \rceil$ as defined above.

1: Iteration 1:

(a) Let $\mathcal{C} = \{\emptyset \neq S \subsetneq V \ : \ \sum_{e\in E_{cur}\cap\delta(S)} u_e + \sum_{e\in E_{T-1}\cap\delta(S)} 2u_e x^*_e < k\}$.

(b) Apply the approximation algorithm for Cover Small Cuts to select edges from $E - E_{T-1} - E_{cur}$ to cover the cuts in $\mathcal{C}$. Add the selected edges to E_{cur}.

(c) Repeat (a), (b) once.

2: For $i = 2, ..., T-1$, Iteration i:

(a) For $\ell = 1, ..., \lfloor (k - 2^{T-i+1})/2^{T-i} \rfloor$:

(i) Let $\mathcal{C} = \{\emptyset \neq S \subsetneq V \ : \ \sum_{e\in E_{cur}\cap\delta(S)} u_e + \sum_{e\in E_{T-i}\cap\delta(S)} 2u_e x^*_e < \ell\, 2^{T-i}\}$.

(ii) Apply the approximation algorithm for Cover Small Cuts to select edges from $E_{T-i+1} - E_{T-i} - E_{cur}$ to cover the cuts in $\mathcal{C}$. Add the selected edges to E_{cur}.

(b) Let $\mathcal{C} = \{\emptyset \neq S \subsetneq V \ : \ \sum_{e\in E_{cur}\cap\delta(S)} u_e + \sum_{e\in E_{T-i}\cap\delta(S)} 2u_e x^*_e < k\}$.

(c) Apply the approximation algorithm for Cover Small Cuts to selected edges from $E - E_{T-i} - E_{cur}$ to cover the cuts in $\mathcal{C}$. Add the selected edges to E_{cur}.

(d) Repeat (b), (c) **two** additional times.

3: Iteration T:

(a) At this point, we have that $\sum_{e\in E_{cur}\cap\delta(S)} u_e + \sum_{e\in E_1\cap\delta(S)} 2u_e x^*_e \geq k$ for all $S \subsetneq V, S \neq \emptyset$. Apply Jain's iterative rounding method to round the edges in E_1 to an integer solution E^*_1, such that $E_{cur} \cup E^*_1$ is a feasible solution to Cap-k-ECSS.

4: Return $E_{cur} \cup E^*_1$.

Iteration 1 In this iteration, we consider the family of small cuts $S \subsetneq V$ where

$$\sum_{e\in E_{cur}\cap\delta(S)} u_e + \sum_{e\in E_{T-1}\cap\delta(S)} 2\ u_e x^*_e < k \qquad \text{(definition of small cuts)}$$

We will cover these cuts using edges in $E - E_{T-1} - E_{cur}$. Note that the definition of small cuts above implies that $\sum_{e\in E_{T-1}\cap\delta(S)} u_e x^*_e < R/2$ where $R = k - \sum_{e\in E_{cur}\cap\delta(S)} u_e$. The knapsack-cover inequality then implies that $\sum_{e\in(E-E_{T-1}-E_{cur})\cap\delta(S)} R x^*_e > R/2$. Thus $2\,x^*_{E-E_{T-1}-E_{cur}}$ is feasible for the Cover Small Cuts problem implying that we incur a cost of at most $5 \cdot 2 \cdot c(x^*_{E_T - E_{T-1}})$ here. We add these edges to E_{cur} and we repeat one more time, i.e. we again consider all cuts where $\sum_{e\in E_{cur}\cap\delta(S)} u_e + \sum_{e\in E_{T-1}\cap\delta(S)} 2u_e x^*_e < k$ and augment these cuts using edges in $E - E_{T-1} - E_{cur}$, incurring a further cost of $5 \cdot 2 \cdot c(x^*_{E-E_{T-1}})$. Now, we will have necessarily satisfied invariant 3 at the end of this iteration. Since, if this invariant were not satisfied for some cut, then this cut participated as a small cut in both instances of the Cover Small Cuts considered in this step. This means we would have added at least two edges across this cut, each of capacity at least $k/2$, ensuring that invariant 3 holds.

Iteration i Phase 1: (Step 2 (a) in Algorithm 2)

We are starting with invariant 1 at the beginning of this iteration (as this corresponds to the invariant 3 that holds at the end of the previous iteration). Hence we have $E_{cur} \cap E_{T-i+1} = \emptyset$ and $\sum_{e\in E_{cur}} u_e + \sum_{e\in E_{T-i+1}} 2u_e x^*_e \geq k$. We will run multiple sub-iterations within this phase. Let's describe the first sub-iteration.

Consider the family of small cuts $S \subsetneq V$ where $\sum_{e\in E_{cur}\cap\delta(S)} u_e + \sum_{e\in E_{T-i}\cap\delta(S)} 2u_e x^*_e < 2^{T-i}$. We will cover these cuts using edges in $E_{T-i+1} - E_{T-i} - E_{cur}$. For these small cuts, we must have

$$\sum_{e \in (E_{T-i+1} - E_{T-i} - E_{cur}) \cap \delta(S)} 2u_e x_e^* \geq k - 2^{T-i},$$

since invariant 1 is true. Note that $u_e \leq 2^{T-i+1}$ for all edges in E_{T-i+1} and so $\sum_{e \in (E_{T-i+1} - E_{T-i} - E_{cur}) \cap \delta(S)} 2x_e^* \geq (k - 2^{T-i})/2^{T-i+1}$. Thus, $2x^*_{E_{T-i+1} - E_{T-i} - E_{cur}} \cdot 2^{T-i+1}/(k - 2^{T-i})$ is feasible for the Cover Small Cuts problem. We add the (approximate) solution of the Cover Small Cut problem to E_{cur}. Note that after this, there are no cuts $S \subsetneq V$ with $\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i} \cap \delta(S)} 2u_e x_e^* < 2^{T-i}$ since we would have added an edge from $E_{T-i+1} - E_{T-i} - E_{cur}$ to E_{cur} and all these edges have capacity at least 2^{T-i}. We now shift the threshold in the definition for Small Cuts to $2 \cdot 2^{T-i}$, and then to $3 \cdot 2^{T-i}$, all the way until $\hat{\ell} \cdot 2^{T-i}$ where $\hat{\ell} = \lfloor (k - 2^{T-i+1})/2^{T-i} \rfloor$. This would imply that $\hat{\ell} \cdot 2^{T-i} \geq k - 2^{T-i+1} - 2^{T-i}$. We describe these sub-iterations in more detail now.

For $\ell = 1, 2, \ldots, \hat{\ell}$, consider the family of small cuts $S \subsetneq V$ where

$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i} \cap \delta(S)} 2u_e x_e^* < \ell \cdot 2^{T-i} \qquad \text{(definition of small cuts)}$$

Since invariant 1 is true (and we have only increased the LHS of invariant 1 during the phase), it must be true that

$$\sum_{e \in (E_{T-i+1} - E_{T-i} - E_{cur}) \cap \delta(S)} 2u_e x_e^* \geq k - \ell \cdot 2^{T-i}$$

Since $u_e \leq 2^{T-i+1}$ for all edges in E_{T-i+1}, we must have

$$\sum_{e \in (E_{T-i+1} - E_{T-i} - E_{cur}) \cap \delta(S)} 2x_e^* \geq (k - \ell \cdot 2^{T-i})/2^{T-i+1}$$

Thus, $2x^*_{E_{T-i+1} - E_{T-i} - E_{cur}} \cdot 2^{T-i+1}/(k - \ell \cdot 2^{T-i})$ is feasible for our instance of the Cover Small Cuts problem. We add an (approximate) solution of this Cover Small Cuts problem to E_{cur}, and move on to the next sub-iteration. At the end of all the sub-iterations, it must be true that

$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i} \cap \delta(S)} 2u_e x_e^* \geq (\hat{\ell}) 2^{T-i} \geq k - 2^{T-i+1} - 2^{T-i}$$

and so invariant 2 is maintained. Let us analyze the cost we incurred in this phase.

The cost we incur is at most $5 \cdot 2 \cdot c(x^*_{E_{T-i+1} - E_{T-i}}) 2^{T-i+1} (\frac{1}{k - 2^{T-i}} + \frac{1}{k - 2 \cdot 2^{T-i}} + \cdots + \frac{1}{k - \hat{\ell} 2^{T-i}})$. We bound this last sum as follows. Note that $\hat{\ell} 2^{T-i} \leq k - 2^{T-i+1}$ and so $k - \hat{\ell} 2^{T-i} \geq 2^{T-i+1}$.

$$\begin{aligned}
&\frac{1}{k - 2^{T-i}} + \frac{1}{k - 2 \cdot 2^{T-i}} + \cdots + \frac{1}{k - \hat{\ell} 2^{T-i}} \\
&= \frac{1}{k - \hat{\ell} 2^{T-i}} + \frac{1}{k - \hat{\ell} 2^{T-i} + 2^{T-i}} + \frac{1}{k - \hat{\ell} 2^{T-i} + 2 \cdot 2^{T-i}} + \cdots \frac{1}{k - \hat{\ell} 2^{T-i} + (\hat{\ell} - 1) \cdot 2^{T-i}} \\
&\leq \frac{1}{2^{T-i+1}} + \sum_{\theta=1}^{\hat{\ell}-1} \frac{1}{k - \hat{\ell} 2^{T-i} + 2^{T-i} \theta} \\
&\leq \frac{1}{2^{T-i+1}} + \int_0^{\hat{\ell}-1} \frac{1}{k - \hat{\ell} 2^{T-i} + 2^{T-i} \theta} d(\theta) \\
&= \frac{1}{2^{T-i+1}} + \frac{1}{2^{T-i}} \Big(\log(k - 2^{T-i}) - \log(k - \hat{\ell} 2^{T-i}) \Big) \\
&= O(\frac{\log k}{2^{T-i}})
\end{aligned}$$

Thus, the total cost we incur in this phase is at most $(5)(2)(2^{T-i+1}/2^{T-i}) O(\log k) c(x^*_{E_{T-i+1} - E_{T-i}})$.

Iteration i Phase 2: (Step 2 (b)-(d) in Algorithm 2)

We are beginning with invariant 2, which is valid at the end of phase 1, and thus for all cuts $S \subsetneq V$, we have

$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i} \cap \delta(S)} 2u_e x_e^* \geq k - 2^{T-i+1} - 2^{T-i}$$

We will add more edges from $E - E_{T-i} - E_{cur}$ to these cuts, if needed, to increase the connectivity to k. Note that all edges in $E - E_{T-i} - E_{cur}$ have capacity at least 2^{T-i} and so at most three more edges need to be added. To do so, we employ the same technique as iteration 1.

Consider the family of small cuts $S \subsetneq V$ where

$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i} \cap \delta(S)} 2u_e x_e^* < k$$

Then, by the knapsack-cover inequalities, $2\, x^*_{E-E_{T-i}-E_{cur}}$ is feasible for the Cover Small Cuts instance. Indeed for these small cuts, we have $\sum_{e \in E_{T-i} \cap \delta(S)} u_e x_e^* < R/2$ where $R = k - \sum_{e \in E_{cur}} u_e$. The knapsack-cover inequality then implies that $\sum_{e \in (E-E_{T-i}-E_{cur}) \cap \delta(S)} R x_e^* > R/2$. Hence, we can solve this Cover Small Cuts instance incurring a cost of at most $10c(x^*_{E-E_{T-i}-E_{cur}})$.

We do so thrice, adding the approximate solution of the Cover Small Cuts instance to E_{cur} and incur a cost of at most $3 \times 10c(x^*)$. At the end of this phase, it must be true that for any cut $S \subsetneq V$, we have

$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_{T-i} \cap \delta(S)} 2u_e x_e^* \geq k.$$

This is precisely invariant 3 and we have completed this phase.

Iteration T In the beginning of the last iteration, we have for any cut $\emptyset \subsetneq S \subsetneq V$:

$$\sum_{e \in E_{cur} \cap \delta(S)} u_e + \sum_{e \in E_1 \cap \delta(S)} 2u_e x_e^* \geq k.$$

But now, we can employ Jain's iterative rounding framework to round the edges in E_1, incurring a cost of at most $(2 \max\{u_e : e \in E_1\}) c(x^*_{E_1}) = 4c(x^*_{E_1})$.

Solving the LP Relaxation: Given a feasible optimal solution x^* to (KCLP: CapkECSS), our rounding algorithm is easily seen to run in polynomial-time. However, it remains to show that we can compute the desired solution x^* in polynomial-time. As before, we would like solve the above LP using the ellipsoid method, but we unfortunately do not know a polynomial-time separation oracle for the entire set of knapsack-cover inequalities. We will instead *iteratively* identify a subset of edges A, and a polynomial-time computable collection of cuts such that as long as knapsack-cover inequalities hold for this collection, we will be able to execute the rounding algorithm. In what follows, we assume w.l.o.g. that the optimal LP solution cost, say $\mathrm{LP}_{\mathrm{opt}}$, is known as it can be identified by running binary search. We can thus replace the minimization objective with simply a feasibility constraint on our solution, namely $\sum_{e \in E} c_e x_e \leq \mathrm{LP}_{\mathrm{opt}}$.

Lemma 13. *There is a polynomial-time algorithm that computes a solution x^* to* (KCLP: CapkECSS) *of value at most LP_{opt} such that for every iteration $i \in [1..(T-1)]$, the solution x^* satisfies the property that $2x^*_{E-E_{T-i}-E_{cur}}$ is feasible for the Cover Small Cuts instances created in steps 1(b), and 2(c) of Algorithm 2.*

Proof. Given a solution x^*, we first check that $\sum_{e \in E} c_e x_e^* \leq \mathrm{LP}_{\mathrm{opt}}$. If not, we return this as a violated constraint. Otherwise, let $\hat{G}(\hat{V}, \hat{E})$ be the capacitated graph where $\hat{V} = V, \hat{E} = E$, and each edge $e \in \hat{E}$ is assigned a capacity of $u_e x_e^*$. We can now check that the capacity of a minimum-cut in $\hat{G}$ is at least k using a polynomial-time global minimum cut algorithm [18]. If not, we return a global minimum cut in $\hat{G}$ as a violated constraint.

By Karger's result [15], we know that there are at most $O(n^4)$ cuts of capacity at most $2k$ (i.e., at most twice the capacity of a minimum-cut), and, moreover, we can enumerate all such cuts of $\hat{G}$ in polynomial time [16]. By iterating over each of the $O(n^4)$ cuts, we can then verify in polynomial-time that the knapsack-cover inequalities are satisfied w.r.t. the set $A = E_{cur}$ in each of the steps 1(b) and 2(c) for cuts whose capacity is at most $2k$. If not, we have found a violated constraint. It remains then to handle the case when we are at step 1(b) or 2(c), and we have a small cut $S \subsetneq V$ such that $\sum_{e\in\delta(S)} u_e x^*_e > 2k$.

In this case, we note that in step 1(b), by the definition of small cuts, we have $\sum_{e\in E_{T-1}\cap\delta(S)} u_e x^*_e < R/2$ where $R = k - \sum_{e\in E_{cur}\cap\delta(S)} u_e$. But then since the total capacity of this cut is at least $2k$, it follows that $\sum_{e\in(E-E_{T-1}-E_{cur})\cap\delta(S)} u_e x^*_e > k$. Since $u_e \le k$ for every edge $e \in E$, it follows that $\sum_{e\in(E-E_{T-1}-E_{cur})\cap\delta(S)} x^*_e > 1$. Thus $2\,x^*_{E-E_{T-1}-E_{cur}}$ is feasible on this cut for the Cover Small Cuts instance. A similar argument can be used to show that in step 2(c), if we have a small cut $S \subsetneq V$ with $\sum_{e\in\delta(S)} u_e x^*_e > 2k$, then $2\,x^*_{E-E_{T-i}-E_{cur}}$ is feasible for the Cover Small Cuts instance. Specifically, by the definition of small cuts, we have $\sum_{e\in E_{cur}\cap\delta(S)} u_e + \sum_{e\in E_{T-i}\cap\delta(S)} 2u_e x^*_e < k$. As the total capacity is at least $2k$, it follows that $\sum_{e\in(E-E_{T-i}-E_{cur})\cap\delta(S)} u_e x^*_e > k$, and since $u_e \le k$ for every edge $e \in E$, $\sum_{e\in(E-E_{T-i}-E_{cur})\cap\delta(S)} x^*_e > 1$. Thus $2x^*_{E-E_{T-i}-E_{cur}}$ is feasible on this cut for the Cover Small Cuts instance.

Finally, if at any step of the rounding algorithm, we identify a violated constraint, then we *re-start* the rounding algorithm from the *very beginning*. It is worth highlighting that the verification of knapsack-cover inequality constraints identified in steps 1(b) and 2(c) of the algorithm, is always done with respect to the solution x^* given by the Ellipsoid algorithm (without any modification). As the rounding progresses, the only thing that changes is the definition of the set $A = E_{cur}$ with respect to which we verify the knapsack-cover constraints. So whenever a violated constraint is identified, it contributes to the iteration count of the ellipsoid algorithm. Since the ellipsoid algorithm terminates after $n^{O(1)}$ iterations of feasibility verification [13], it must be the case that after at most $n^{O(1)}$ re-starts of the rounding process, we arrive at a solution x^* to (KCLP: CapkECSS) of value at most $\mathrm{LP}_{\mathrm{opt}}$ such that the solution satisfies the property that $2x^*_{E-E_{T-i}-E_{cur}}$ is feasible for the Cover Small Cuts instances created in steps 1(b) and 2(c).

$\square$

Theorem 14. *Let* OPT *be the optimal solution value for a given instance of Cap-k-ECSS. Then there is a polynomial-time algorithm that computes a solution x^* to* (KCLP: CapkECSS) *of value at most* OPT *(possibly satisfying only a subset of the constraints) and rounds it to obtain a feasible integer solution of cost at most* $O(\log k) \cdot \mathrm{OPT}$

Appendix A Formulating (p,q)-FGC as a Cap-k-ECSS Problem

We attempt to model (p,q)-FGC as the Cap-k-ECSS problem. Let us take p and q to be parameters. Our goal is to find conditions on p and q such that (p,q)-FGC can be formulated as a Cap-k-ECSS problem. We show that one can formulate a (p,q)-FGC problem as an equivalent Cap-k-ECSS problem if and only if $p=1$ or $q=1$.

We make the following assumptions:

- p and q are positive integers;
- each unsafe edge is assigned the capacity $u(p,q)>0$;
- each safe edge is assigned the capacity $s(p,q) \geq u(p,q) > 0$;
- the requirement k of the Cap-k-ECSS problem is fixed at $p \cdot s(p,q) = (p+q) \cdot u(p,q)$, because each nontrivial cut of (p,q)-FGC is required to have either p safe edges or $p+q$ edges.

Clearly, a cut that violates the requirement of (p,q)-FGC should have capacity less than k. Let us call this property (0).

For any $i = 1, 2, ..., p-1$, a cut does not satisfy the requirement of (p,q)-FGC if it has $\leq p-i$ safe edges and $\leq q+i-1$ unsafe edges.

This gives the constraint $(p-i)s(p,q) + (q+i-1)u(p,q) < k$.

Since $k = p \cdot s(p,q) = (p+q) \cdot u(p,q)$, we have $s(p,q) = u(p,q)(p+q)/p$. Starting from property (0), we get the following (each line follows from the preceding line):

$$
\begin{aligned}
(p-i)s(p,q) + (q+i-1)u(p,q) &< k = (p)s(p,q) \\
(-i)s(p,q) + (q+i-1)u(p,q) &< 0 \\
(-i)u(p,q)\frac{p+q}{p} + (q+i-1)u(p,q) &< 0 \\
\text{Since } u(p,q) > 0, \text{ this is equivalent to} & \\
q+i-1 &< (i)\frac{p+q}{p} \\
pq + ip - p &< (i)(p+q) \\
pq &< iq + p \qquad \text{(inequality } (*))
\end{aligned}
$$

Recall that $i \in \{1, ..., p-1\}$. Suppose that $i=1$ and assume that $q \geq p$. Then inequality $(*)$ is the same as $p < 1 + \dfrac{p}{q} \leq 2$, that is, $p \leq 1$. Thus, inequality $(*)$ implies either $q < p$ or $p = 1$.

Similarly, by taking $i = 1$ and assuming that $p \geq q$, inequality $(*)$ can be written as $q < 1 + \dfrac{q}{p} \leq 2$, that is, $q \leq 1$. Thus, inequality $(*)$ implies either $p < q$ or $q = 1$.

Hence, either $p=1$ or $q=1$ (since $p<q$ and $q<p$ cannot both be true).

Appendix B An $O(p \log n)$-Approximation Algorithm for (p,q)-FGC via Covering Integer Programs

We apply a theorem from Chekuri & Quanrud [9] that gives an approximation algorithm for Covering Integer Programs (abbreviated as CIPs). We recap Theorem 2.3 from [9].

Theorem [Chekuri & Quanrud [9]] *Given $A \in [0,1]^{m \times n}$, $c \in \mathbb{R}^n_{\geq 0}$, $d \in \mathbb{Z}^n_{\geq 0}$, $x \in \mathbb{R}^n_{\geq 0}$, such that x satisfies the Knapsack Cover Inequalities. Let the approximation parameter be $\alpha = \ln \Delta_0 + \ln \ln \Delta_0 + O(1)$. The algorithm runs in time $\|A\|_0$ and finds an integer vector z of cost $\leq \alpha \cdot c^\top x$ such that $Az \geq 1$ and $0 \leq z \leq d$. (Here, $\|A\|_0$*

denotes the number of nonzeros of A, and Δ_0 denotes the maximum number of nonzeros in any column of A.) (Note: for our application to (p,q)-FGC, we take $\Delta_0 = |V(G)| = \hat{n}$.)

We apply the theorem from [9] to cover the deficient cuts while augmenting a solution to (i,q)-FGC to a solution to $(i+1,q)$-FGC. We have two types of deficient cuts:

(1) Cuts with exactly $(i+q)$ edges (safe or unsafe); we need to pick one more edge (safe or unsafe) to cover each such cut.

(2) Cuts with exactly i safe edges and j unsafe edges, where $j \in \{0,1,2,\ldots,q-1\}$; for each such cut, we need to pick either one safe edge or $(q+1-j)$ unsafe edges; we can formulate this requirement as a valid inequality constraint of a CIP such that we can apply the theorem of [9].

(Note: if a cut has i safe edges and q unsafe edges, then it is a cut of type (1)).

The algorithm starts by computing a (1,q)-FGC solution of cost $O(\log n)$OPT. In detail, we formulate (precisely) the (1,q)-FGC problem as a Capacitated Network Design (CND) problem, by assigning capacities of $(q+1)$ and one, respectively, to the safe edges and the unsafe edges. Then, we apply the $O(\log n)$ approximation algorithm of Chakrabarty et al. [7] to our CND problem.

Next, we apply $p-1$ iterations. Iteration i starts with a solution to (i,q)-FGC and augments deficient cuts (if any) to obtain a solution to $(i+1,q)$-FGC. Each iteration formulates a CIP and finds a solution to this CIP of cost $O(\log n)$OPT, using the algorithm/theorem of [9].

Thus, the overall approximation ratio is $O(p \log n)$.

We present the details for iteration i in what follows. Let H be the solution to (i,q)-FGC, at the start of the iteration. Our goal is to find all the deficient cuts, then write down a CIP, then find an approximately optimal solution to the CIP via [9, Theorem 2.3]. We start by formulating a CND using the following parameters:

capacity of a safe edge:	$i+q$
capacity of an unsafe edge:	i
required capacity (for each non-trivial cut):	$i(i+q)$

Observe that the CND graph (corresponding to H) has capacity $\geq i(i+q)$ for every non-trivial cut, because H is a solution to (i,q)-FGC.

Claim 15. *If a cut of the CND graph has capacity $\geq 2(i+1)(i+q)$, then this cut satisfies the requirement of $(i+1,q)$-FGC.*

Proof. Suppose safe edges contribute at least half the capacity of this cut; then this cut has $\geq i+1$ safe edges. Otherwise, unsafe edges contribute at least half the capacity of this cut; then this cut has $\geq (i+1)(i+q)/i \geq i+q+1$ unsafe edges. ☐

We apply the algorithm of Nagamochi, Nishimura, & Ibaraki [16] to list all the cuts in the CND graph with capacities in the range $[i(i+q),\ 2(i+1)(i+q)]$ in (deterministic) polynomial time. Note that $2(i+1)(i+q) \leq 4i(i+q)$, hence, by Karger's results, the number of cuts in this range is $O(n^8)$.

Finally, we set up the constraints matrix A for the CIP. For each deficient cut $\delta(S_k)$ (in our list of cuts of the CND graph), we write a constraint of the form $\text{row}_k(A) \cdot x \geq b_k$. For a deficient cut of type (1), i.e., a cut with $i+q$ edges, we have the constraint $\sum_{e \in \delta(S_k) \cap (E(G)-E(H))} x_e \geq 1$.

Next, for notational convenience, define $a_j = (q+1-j)$. For a deficient cut of type (2), i.e., a cut with i safe edges and $\leq q-1$ unsafe edges, let $j \in \{0,\ldots,q-1\}$ denote the number of unsafe edges; for each unsafe edge $e \in E(G)-E(H)$ we fix the coefficient $A_{k,e} = 1$, and for each safe edge $f \in E(G)-E(H)$ we fix the coefficient $A_{k,f} = a_j$, and we fix the RHS coefficient $b_k = a_j$. Thus, we have the constraint

$$a_j \sum_{f \in \delta(S_k) \cap \mathcal{S} \cap (E(G)-E(H))} x_f + \sum_{e \in \delta(S_k) \cap \mathcal{U} \cap (E(G)-E(H))} x_e \geq a_j.$$

References

[1] David Adjiashvili, Felix Hommelsheim, and Moritz Mühlenthaler. Flexible Graph Connectivity. *Mathematical Programming*, 192:409–441, 2022. doi:10.1007/s10107-021-01664-9.

[2] Ishan Bansal. A global analysis of the primal-dual method for pliable families. *CoRR*, abs/2308.15714, 2024. URL: https://arxiv.org/abs/2308.15714v2, doi:10.48550/ARXIV.2308.15714.

[3] Ishan Bansal, Joseph Cheriyan, Logan Grout, and Sharat Ibrahimpur. Improved Approximation Algorithms by Generalizing the Primal-Dual Method Beyond Uncrossable Functions. *CoRR*, abs/2209.11209v2, 2022. URL: https://arxiv.org/abs/2209.11209v2.

[4] Ishan Bansal, Joseph Cheriyan, Logan Grout, and Sharat Ibrahimpur. Improved approximation algorithms by generalizing the primal-dual method beyond uncrossable functions. *Algorithmica*, 86(8):2575–2604, 2024. URL: https://doi.org/10.1007/s00453-024-01235-2, doi:10.1007/S00453-024-01235-2.

[5] Sylvia C. Boyd, Joseph Cheriyan, Arash Haddadan, and Sharat Ibrahimpur. Approximation algorithms for flexible graph connectivity. *Mathematical Programming*, 2023. doi:10.1007/s10107-023-01961-5.

[6] Robert D. Carr, Lisa K. Fleischer, Vitus J. Leung, and Cynthia A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, page 106–115, USA, 2000. Society for Industrial and Applied Mathematics.

[7] 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.

[8] Chandra Chekuri and Rhea Jain. Approximation Algorithms for Network Design in Non-Uniform Fault Models. In *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming*, volume 261, article 36, pages 1–20, 2023. doi:10.4230/LIPIcs.ICALP.2023.36.

[9] Chandra Chekuri and Kent Quanrud. On approximating (sparse) covering integer programs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1596–1615. SIAM, 2019. URL: https://doi.org/10.1137/1.9781611975482.97, doi:10.1137/1.9781611975482.97.

[10] Michael Dinitz, Ama Koranteng, and Guy Kortsarz. Relative survivable network design. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPIcs*, pages 41:1–41:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2022.41, doi:10.4230/LIPICS.APPROX/RANDOM.2022.41.

[11] Michael Dinitz, Ama Koranteng, Guy Kortsarz, and Zeev Nutov. Improved approximations for relative survivable network design. In Jaroslaw Byrka and Andreas Wiese, editors, *Approximation and Online Algorithms - 21st International Workshop, WAOA 2023, Amsterdam, The Netherlands, September 7-8, 2023, Proceedings*, volume 14297 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2023. URL: https://doi.org/10.1007/978-3-031-49815-2_14, doi:10.1007/978-3-031-49815-2_14.

[12] 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.

[13] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer Berlin, 1993. doi:10.1007/978-3-642-78240-4.

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

[15] David R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm. In Vijaya Ramachandran, editor, *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas, USA*, pages 21–30. ACM/SIAM, 1993. URL: http://dl.acm.org/citation.cfm?id=313559.313605.

[16] Hiroshi Nagamochi, Kazuhiro Nishimura, and Toshihide Ibaraki. Computing All Small Cuts in an Undirected Network. *SIAM Journal on Discrete Mathematics*, 10(3):469–481, 1997. doi:10.1137/S0895480194271323.

[17] Zeev Nutov. Improved approximation ratio for covering pliable set families. *CoRR*, 2024. URL: https://arxiv.org/abs/2404.00683, arXiv:2404.00683, doi:https://doi.org/10.48550/arXiv.2404.00683.

[18] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, Berlin Heidelberg New York, 2003.

[19] 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.