

Improved Approximation Algorithms by Generalizing the Primal-Dual Method Beyond Uncrossable Functions

Ishan Bansal¹, Joseph Cheriyan^{2*}, Logan Grout¹, Sharat Irahimpur³

¹Cornell University, Operations Research and Information Engineering, Ithaca, NY, USA.

²Department of Combinatorics and Optimization, University of Waterloo, Canada.

³Department of Mathematics, London School of Economics and Political Science, UK.

*Corresponding author(s). E-mail(s): jcheriyan@uwaterloo.ca;

Contributing authors: ib332@cornell.edu; logancgrout@gmail.com, lbg58@cornell.edu; S.Irahimpur@lse.ac.uk;

Abstract

We address long-standing open questions raised by Williamson, Goemans, Vazirani and Mihail pertaining to the design of approximation algorithms for problems in network design via the primal-dual method (Combinatorica 15(3):435-454, 1995). Williamson et al. prove an approximation ratio of two for connectivity augmentation problems where the connectivity requirements can be specified by uncrossable functions. They state: “Extending our algorithm to handle non-uncrossable functions remains a challenging open problem. The key feature of uncrossable functions is that there exists an optimal dual solution which is laminar . . . A larger open issue is to explore further the power of the primal-dual approach for obtaining approximation algorithms for other combinatorial optimization problems.”

Our main result proves that the primal-dual algorithm of Williamson et al. achieves an approximation ratio of **16** for a class of functions that generalizes the notion of an uncrossable function. There exist instances that can be handled by our methods where none of the optimal dual solutions has a laminar support.

We present three applications of our main result to problems in the area of network design.

- (1) A **16**-approximation algorithm for augmenting a family of small cuts of a graph G . The previous best approximation ratio was $O(\log |V(G)|)$.
- (2) A $16 \cdot \lceil k/u_{\min} \rceil$ -approximation algorithm for the Cap- k -ECSS problem which is as follows: Given an undirected graph $G = (V, E)$ with edge costs $c \in \mathbb{Q}_{\geq 0}^E$ and edge capacities $u \in \mathbb{Z}_{\geq 0}^E$, 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 ; u_{\min} (respectively, u_{\max}) denotes the minimum (respectively, maximum) capacity of an edge in E , and w.l.o.g. $u_{\max} \leq k$. The previous best approximation ratio was $\min(O(\log |V|), k, 2u_{\max})$.

(3) A **20**-approximation algorithm for the model of $(p, 2)$ -Flexible Graph Connectivity. The previous best approximation ratio was $O(\log |V(G)|)$, where G denotes the input graph.

Keywords: Approximation algorithms, Edge-connectivity of graphs, f-Connectivity problem, Flexible graph connectivity, Minimum cuts, Network design, Primal-dual method, Small cuts

1 Introduction

The primal-dual method is a well-known algorithmic discovery of the past century. Kuhn (1955) [1] presented a primal-dual algorithm for weighted bipartite matching, and Dantzig et al. (1957) [2] presented a generalization for solving linear programs. Primal-dual methods for problems in combinatorial optimization are based on linear programming (LP) relaxations; the associated linear programs (LPs) are crucial for the design and analysis of these algorithms. A key feature of the primal-dual method is that it leads to self-contained combinatorial algorithms that do not require solving the underlying LPs. This key feature makes it attractive for both theoretical studies and real-world computations.

Several decades after the pioneering work of Kuhn, the design of approximation algorithms for NP-hard problems emerged as an important area of research. Agrawal, Klein and Ravi [3] designed and analyzed a primal-dual approximation algorithm for the Steiner forest problem. Goemans and Williamson [4] generalized this algorithm to constrained forest problems. Subsequently, Williamson, Goemans, Vazirani and Mihail [5] (abbreviated WGMV) extended the methods of [4] to obtain a primal-dual 2-approximation algorithm for the problem of augmenting the connectivity of a graph to satisfy requirements specified by uncrossable functions. These functions are versatile tools for modeling several key problems in the area of network design.

Network design encompasses diverse problems that find applications in sectors like transportation, facility location, information security, and resource connectivity, to name a few. The area has been studied for decades and it has led to major algorithmic innovations. Most network-design problems are NP-Hard, and many of these problems are APX-hard. Consequently, research in the area has focused on the design and analysis of good approximation algorithms, preferably with a small constant-factor approximation ratio. In the context of network design, many of the $O(1)$ approximation algorithms rely on a particular property called *uncrossability*, see the books by Lau, Ravi & Singh [6], Vazirani [7], and Williamson & Shmoys [8]. This property has been leveraged in various ways to obtain algorithms with $O(1)$ approximation ratios for problems such as survivable network design [9], min-cost/min-size k -edge connected spanning subgraph [10, 11], min-cost 2-node connected spanning subgraph [12], $(p, 1)$ -flexible graph connectivity [13], etc. The primal-dual method is one of the most successful algorithmic paradigms that leverages these uncrossability properties. On the other hand, when the uncrossability property does not hold, then most of the known techniques in this area for designing $O(1)$ approximation algorithms fail to work. WGMV [5] conclude their paper with the following remark:

Extending our algorithm to handle non-uncrossable functions remains a challenging open problem. The key feature of uncrossable functions is that there exists an optimal dual solution which is laminar ... A larger open issue is to explore further the power of the primal-dual approach for obtaining approximation algorithms for other combinatorial optimization problems. Handling all non-uncrossable functions is ruled out by the fact that there exist instances corresponding to non-uncrossable $\{0, 1\}$ functions whose relative duality gap is larger than any constant.

Our main contribution is an extension of the WGMV primal-dual algorithm and its analysis to a class of functions that is more general than the class of uncrossable functions. Our main result is an approximation ratio of 16 for the larger class of functions. We apply our main result to give improved approximation ratios for three problems: (i) augmenting all small cuts of a graph, (ii) finding a minimum-cost capacitated k -edge connected subgraph, and (iii) the $(p, 2)$ -FGC problem. We give a detailed discussion of our results in Sections 1.2–1.5.

Primal-dual algorithms for solving network design problems work iteratively, starting with a graph that has no edges, and adding a set of edges in each iteration, until the set of picked edges forms a feasible subgraph (i.e., a subgraph satisfying the requirements of the problem). In our context, the problem requirements are formalized by specifying a connectivity requirement (which is a non-negative integer) for each set of nodes. Consider one of the iterations. Let F denote the set of edges picked until the start of this iteration. A set of nodes S is said to be *violated* if the number of F -edges in the cut of S is less than the pre-specified connectivity requirement of S . An edge is deemed to be useful if it is in the cut of a violated set S . The goal of the iteration is to buy a cheap set of useful edges.

Clearly, the family of violated sets is important for the design and analysis of these algorithms. (Recall that sets $A, B \subseteq V$ are said to *cross* if each of the four sets $A \cap B$, $V \setminus (A \cup B)$, $A \setminus B$, $B \setminus A$ is nonempty.) A family \mathcal{F} of sets is called *uncrossable* if the following holds:

$$A, B \in \mathcal{F} \implies A \cap B, A \cup B \in \mathcal{F} \text{ or } A \setminus B, B \setminus A \in \mathcal{F}.$$

Informally speaking, the uncrossability property of \mathcal{F} ensures that the (inclusion-wise) minimal sets of \mathcal{F} can be considered independently. Now, suppose that \mathcal{F} is the family of violated sets at some iteration of the primal-dual algorithm, and suppose that \mathcal{F} is uncrossable. Then observe that an (inclusion-wise) minimal violated set $A \in \mathcal{F}$ cannot cross another set $S \in \mathcal{F}$; otherwise, we get a contradiction since $A, S \in \mathcal{F}$ implies that a proper subset of A is in \mathcal{F} (either $A \cap S \in \mathcal{F}$ or $A \setminus S \in \mathcal{F}$, because \mathcal{F} is uncrossable). This key property is one of the levers used in the design of some of the $O(1)$ -approximation algorithms for problems in network design.

Unfortunately, there are important problems in network design where the family of violated sets does not form an uncrossable family; for instance, see the discussion in Appendix D. Let us call a family of sets \mathcal{F} *pliable* if the following holds:

$$A, B \in \mathcal{F} \implies \text{at least two of } A \cap B, A \cup B, A \setminus B, B \setminus A \text{ are in } \mathcal{F}.$$

Clearly, an uncrossable family of sets is a pliable family. In Appendix A, we show that the WGMV primal-dual algorithm has a super-constant approximation ratio for pliable families. In more detail, we present an example with n nodes such that the cost of the solution found by the WGMV algorithm is $\Omega(\sqrt{n})$ times the cost of an optimal solution. Nevertheless, we prove an $O(1)$ approximation ratio for the WGMV primal-dual algorithm applied to pliable families satisfying an additional property that we call property (γ) ; the formal definition is given below. Property (γ) allows a minimal violated set to cross another violated set, but, crucially, it does not allow a minimal violated set to cross an arbitrary number of violated sets in arbitrary ways. Thus, pliable families satisfying property (γ) are (strictly) more general than uncrossable families. Informally speaking, this is what allows us to obtain improved approximation ratios for problems such as $(p, 2)$ -FGC via the WGMV primal-dual algorithm. Now, we state property (γ) .

Property (γ) : For any $F \subseteq E$ and for any violated sets (w.r.t. F) C, S_1, S_2 , such that
(ii) $S_1 \subsetneq S_2$, (ii) C is inclusion-wise minimal, and (iii) C crosses both S_1, S_2 ,
 $S_2 \setminus (S_1 \cup C)$ is either empty or violated.

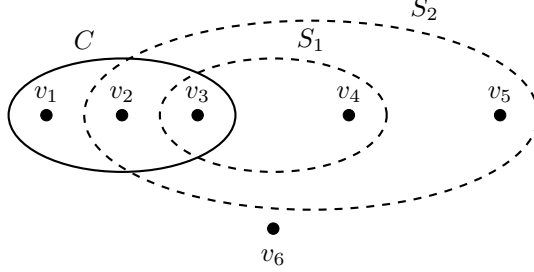


Fig. 1 Illustration of property (γ), and the sets C, S_1, S_2 . The set $S_2 \setminus (S_1 \cup C)$ is non-empty.

1.1 f -connectivity problem

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 $c \in \mathbb{Q}_{\geq 0}^E$ on the edges and a requirement function $f : 2^V \rightarrow \{0, 1\}$ on subsets of nodes. We are interested in finding an edge-set $F \subseteq E$ with minimum cost $c(F) := \sum_{e \in F} c_e$ such that for all cuts $\delta(S)$, $S \subseteq V$, we have $|\delta(S) \cap F| \geq f(S)$. This problem can be formulated as the following integer program where the binary variable x_e models the inclusion of the edge e in F :

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

In its most general form, the f -connectivity problem is hard to approximate within a logarithmic factor. This can be shown via a reduction from the hitting set problem which has a logarithmic hardness-of-approximation threshold; see the discussion at the end of Section 2.1.

If our goal is to obtain $O(1)$ -approximation algorithms, then we must require f to have some properties such that the corresponding f -connectivity problem captures some interesting problems in connectivity augmentation, while ensuring that the f -connectivity problem does not capture problems that have super-constant hardness-of-approximation thresholds. This motivates the following definitions.

Definition 1.1 ([5]). *A function $f : 2^V \rightarrow \{0, 1\}$ satisfying $f(V) = 0$ is called uncrossable if for any $A, B \subseteq V$ with $f(A) = f(B) = 1$, we have $f(A \cap B) = f(A \cup B) = 1$ or $f(A \setminus B) = f(B \setminus A) = 1$.*

Definition 1.2. A function $f : 2^V \rightarrow \{0, 1\}$ satisfying $f(V) = 0$ is called *pliable* if for any $A, B \subseteq V$ with $f(A) = f(B) = 1$, we have $f(A \cap B) + f(A \cup B) + f(A \setminus B) + f(B \setminus A) \geq 2$.

Clearly, the problem of augmenting an uncrossable (respectively, pliable) family of violated sets can be formulated as an f -connectivity problem whose requirement function is an uncrossable (respectively, pliable) function.

1.2 Our main result

Our main result is that the WGMV primal-dual algorithm achieves an approximation ratio of $O(1)$ for the f -connectivity problem whenever f is a pliable function satisfying property (γ) . As discussed above, the analysis of WGMV relies on the property that for any (inclusion-wise) minimal violated set C and any violated set S , either C is a subset of S or C is disjoint from S ([5, Lemma 5.1(3)]). This property does not hold for a pliable function satisfying property (γ) ; see the instance described in Appendix D. Informally speaking, our analysis of the WGMV primal-dual algorithm leverages property (γ) and proves an approximation ratio of $O(1)$; recall that property (γ) does not allow a minimal violated set to cross an arbitrary number of violated sets in arbitrary ways. (See Section 2 for definitions of terms such as violated set w.r.t. f, F .)

Theorem 1.1. Let $G = (V, E)$ be an undirected graph with nonnegative costs $c : E \rightarrow \mathbb{Q}_{\geq 0}$ on its edges, and let $f : 2^V \rightarrow \{0, 1\}$ be a pliable function satisfying property (γ) . Suppose that there is a subroutine that, for any given $F \subseteq E$, computes all minimal violated sets w.r.t. f and F . Then, in polynomial time and using a polynomial number of calls to the subroutine, we can compute a 16-approximate solution to the given instance of the f -connectivity problem.

In the next three sections, we describe problems in the area of network design where Theorem 1.1 gives new/improved approximation algorithms. In each of these applications, we formulate an f -connectivity problem where the function f is a pliable function with property (γ) .

1.3 Application 1: Augmenting a Family of Small Cuts

Our first application is on finding a minimum-cost augmentation of a family of small cuts in a graph. In an instance of the AugSmallCuts problem, we are given an undirected capacitated graph $G = (V, E)$ with edge-capacities $u \in \mathbb{Q}_{\geq 0}^E$, a set of links $L \subseteq \binom{V}{2}$ with costs $c \in \mathbb{Q}_{\geq 0}^L$, and a threshold $\tilde{\lambda} \in \mathbb{Q}_{\geq 0}$. A subset $F \subseteq L$ of links is said to *augment* 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 augments all non-empty $S \subsetneq V$ with $u(\delta_E(S)) < \tilde{\lambda}$.

Some special cases of the AugSmallCuts problem have been studied previously, and, to the best of our knowledge, there is no previous publication on the general version of this problem. Let $\lambda(G)$ denote the minimum capacity of a cut of G , thus, $\lambda(G) := \min\{u(\delta_E(S)) : \emptyset \subsetneq S \subsetneq V\}$. Assuming u is integral and $\tilde{\lambda} = \lambda(G) + 1$, we get the well-known connectivity augmentation problem for which constant-factor approximation algorithms are known [14, 15]. On the other hand, when $\tilde{\lambda} = \infty$, a minimum-cost spanning tree of (V, L) , if one exists, gives an optimal solution to the problem.

Our main result here is an $O(1)$ -approximation algorithm for the AugSmallCuts problem that works for any choice of $\tilde{\lambda}$. The proof of the following theorem is given in Section 4.

Theorem 1.2. There is a 16-approximation algorithm for the AugSmallCuts problem.

We refer the reader to Benczur & Goemans [16] and the references therein for results on the representations of the near-minimum cuts of graphs; they do not study the problem of augmenting the near-minimum cuts.

1.4 Application 2: Capacitated k -Edge-Connected Subgraph Problem

In the capacitated k -edge-connected subgraph problem (Cap- k -ECSS), we are given an undirected graph $G = (V, E)$ with edge costs $c \in \mathbb{Q}_{\geq 0}^E$ and edge capacities $u \in \mathbb{Z}_{\geq 0}^E$. The goal is to find a minimum-cost subset of the edges $F \subseteq E$ such that the capacity across any cut in (V, F) is at least k , i.e., $u(\delta_F(S)) \geq k$ for all non-empty sets $S \subsetneq V$. Let u_{\max} and u_{\min} , respectively, denote the maximum capacity of an edge in E and the minimum capacity of an edge in E . We may assume (w.l.o.g.) that $u_{\max} \leq k$. (There is a “soft” problem related to Cap- k -ECSS where a solution subgraph is allowed to contain multiple copies of any edge of G ; in particular, for any edge e of G and any integer $\mu_e \geq 0$, a solution subgraph could pick μ_e copies of e to obtain a capacity of $\mu_e u_e$ at a cost of $\mu_e c_e$; in this context, note that $\mu_e \in \{0, 1\}, \forall e \in E$, for Cap- k -ECSS. There is a simple $O(1)$ approximation algorithm for this “soft” problem, see [17, Section 1]. This “soft” problem is not relevant for our paper.)

We mention that there are well-known 2-approximation algorithms for the special case of the Cap- k -ECSS problem with $u_{\max} = u_{\min} = 1$, which is the problem of finding a minimum-cost k -edge connected spanning subgraph. Khuller & Vishkin [18] presented a combinatorial 2-approximation algorithm and Jain [9] matched this approximation guarantee via the iterative rounding method.

Goemans et al. [19] gave a $2k$ -approximation algorithm for the general Cap- k -ECSS problem. Chakrabarty et al. [17] gave a randomized $O(\log |V(G)|)$ -approximation algorithm; note that this approximation guarantee is independent of k but does depend on the size of the underlying graph. Recently, Boyd et al. [13] improved on these results by providing a $\min(k, 2u_{\max})$ -approximation algorithm. We present a $(16 \cdot \lceil k/u_{\min} \rceil)$ -approximation algorithm for the Cap- k -ECSS problem. This gives improved approximation ratios when both u_{\min} and u_{\max} are sufficiently large; observe that $(16 \cdot \lceil k/u_{\min} \rceil)$ is less than $\min(k, 2u_{\max})$ when $k \geq u_{\max} \geq u_{\min} \geq 32$ and $u_{\min} \cdot u_{\max} \geq 16k$.

Theorem 1.3. *There is a $16 \cdot \lceil k/u_{\min} \rceil$ -approximation algorithm for the Cap- k -ECSS problem.*

The proof of Theorem 1.3 is given in Section 5.

1.5 Application 3: $(p, 2)$ -Flexible Graph Connectivity

Adjashvili, Hommelsheim and Mühlenhaller [20] introduced the model of Flexible Graph Connectivity that we denote by FGC. Boyd, Cheriyan, Haddadan and Ibrahimpur [13] introduced a generalization of FGC. Let $p \geq 1$ and $q \geq 0$ be integers. In an instance of the (p, q) -Flexible Graph Connectivity problem, denoted (p, q) -FGC, we are given an undirected 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 edge-costs $c \in \mathbb{Q}_{\geq 0}^E$. 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 $(V, F \setminus F')$ is 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. [13] presented a 4-approximation algorithm for $(p, 1)$ -FGC based on the WGMV primal-dual method, 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. Concurrently with our work, Chekuri and Jain [21, 22] obtained $O(p)$ -approximation algorithms for $(p, 2)$ -FGC, $(p, 3)$ -FGC and $(2p, 4)$ -FGC; in particular, for $(p, 2)$ -FGC, they prove an approximation ratio of $(2p + 4)$. Chekuri and Jain [21, 23]

have several other results for problems in the flexible graph connectivity model; they introduce and study non-uniform fault models such as (p, q) -Flex-SNDP.

Our main result here is an $O(1)$ -approximation algorithm for the $(p, 2)$ -FGC problem.

Theorem 1.4. *There is a 20-approximation algorithm for the $(p, 2)$ -FGC problem. Moreover, for even p , the approximation ratio is 6.*

Note that in comparison to [21, 22], Theorem 1.4 yields a better approximation ratio when $p > 8$ or $p \in \{2, 4, 6, 8\}$. For $p = 1$, the approximation ratio of 3 from [13] is better than the guarantees given by [21, 22] and Theorem 1.4. The proof of Theorem 1.4 is given in Section 6.

1.6 Related work

Goemans & Williamson [4] formulated several problems in network design as the f -connectivity problem where f is a proper function. A symmetric function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ with $f(V) = 0$ is said to be *proper* if $f(A \cup B) \leq \max(f(A), f(B))$ for any pair of disjoint sets $A, B \subseteq V$.

Jain [9] designed the iterative rounding framework for the setting when f is weakly supermodular and presented a 2-approximation algorithm. A function f is said to be *weakly supermodular* if $f(A) + f(B) \leq \max(f(A \cap B) + f(A \cup B), f(A \setminus B) + f(B \setminus A))$ for any $A, B \subseteq V$. One can show that proper functions are weakly supermodular, [19, Lemma 2.1]. We mention that an uncrossable function may not be weakly supermodular. If we have a weakly supermodular function f and sets $A, B \subseteq V$ such that $f(A) + f(B) = 1$, then at least one of the four sets $A \cap B, A \cup B, A \setminus B, B \setminus A$ must have f -value at least one, whereas, this condition need not hold for uncrossable functions. See the remark in [13, Section 4] for an example.

Although our paper focuses on theory, let us mention that extensive computational research over decades shows that the primal-dual method works well. In particular, computational studies of some of the well-known primal-dual approximation algorithms have been conducted, and the consensus is that these algorithms work well in practice, see [24, Section 4.9], [25], [26], [27], [28].

After the posting of a preliminary version of our paper [29], Bansal [30] and Nutov [31] have posted related results.

2 Preliminaries

This section has definitions and preliminary results. Our notation and terms are consistent with [32, 33], and readers are referred to those texts for further information.

For a positive integer k , we use $[k]$ to denote the set $\{1, \dots, k\}$. Sets $A, B \subseteq V$ are said to *cross*, denoted $A \bowtie B$, if each of the four sets $A \cap B, V \setminus (A \cup B), A \setminus B, B \setminus A$ is non-empty; on the other hand, if A, B do not cross, then either $A \cup B = V$, or A, B are disjoint, or one of A, B is a subset of the other one. A family of sets $\mathcal{L} \subseteq 2^V$ is said to be *laminar* if for any two sets $A, B \in \mathcal{L}$ either A and B are disjoint or one of them is a subset of the other one.

We may use abbreviations for some standard terms, e.g., we may use “ (p, q) -FGC” as an abbreviation for “the (p, q) -FGC problem”. In some of our discussions, we may use informal phrasing such as “we apply the primal-dual method to augment a pliable function.”

2.1 Graphs, Subgraphs, and Related Notions

Let $G = (V, E)$ be an undirected multi-graph (possibly containing parallel edges but no loops) with non-negative costs $c \in \mathbb{R}_{\geq 0}^E$ on the edges. We take G to be the input graph, and we use n to denote $|V(G)|$. For a set of edges $F \subseteq E(G)$, $c(F) := \sum_{e \in F} c(e)$, and for a subgraph G' of G , $c(G') := c(E(G'))$. For any instance G , we use $\text{OPT}(G)$ to denote the minimum cost of a feasible subgraph (i.e., a subgraph that satisfies the requirements of the problem). When there is no danger of ambiguity, we use OPT rather than $\text{OPT}(G)$.

Let $G = (V, E)$ be any multi-graph, let $A, B \subseteq V$ be two disjoint node-sets, and let $F \subseteq E$ be an edge-set. We denote the multi-set of edges of G with exactly one end-node in each of A and B by $E(A, B)$, thus, $E(A, B) := \{e = uv : u \in A, v \in B\}$. Moreover, we use $\delta_E(A)$ or $\delta(A)$ to denote $E(A, V \setminus A)$; $\delta(A)$ is called a *cut* of G . By a p -cut we mean a cut of size p . We use $G[A]$ to denote the subgraph of G induced by A , $G - A$ to denote the subgraph of G induced by $V \setminus A$, and $G - F$ to denote the graph $(V, E \setminus F)$. We may use relaxed notation for singleton sets, e.g., we may use $G - v$ instead of $G - \{v\}$, etc. A multi-graph G is called k -edge connected if $|V(G)| \geq 2$ and for every $F \subseteq E(G)$ of size $< k$, $G - F$ is connected.

Fact 2.1. *Let $A, B \subseteq V$ be a pair of crossing sets. For any edge-set $F \subseteq \binom{V}{2}$ and any $S \in \{A \cap B, A \cup B, A \setminus B, B \setminus A\}$, we have $\delta_F(S) \subseteq \delta_F(A) \cup \delta_F(B)$.*

Proof. By examining cases, we can show that $e \in \delta_F(S) \implies e \in \delta_F(A)$ or $e \in \delta_F(B)$. \square

For any function $f : 2^V \rightarrow \{0, 1\}$ and any edge-set $F \subseteq E$, we say that $S \subseteq V$ is *violated w.r.t. f* , F if $|\delta_F(S)| < f(S)$, i.e., if $f(S) = 1$ and there are no F -edges in the cut $\delta(S)$. We drop f and F when they are clear from the context. The next observation states that the violated sets w.r.t. any pliable function f and any “augmenting” edge-set F form a pliable family.

Fact 2.2. *Let $f : 2^V \rightarrow \{0, 1\}$ be a pliable function and $F \subseteq E$ be an edge-set. Define the function $f' : 2^V \rightarrow \{0, 1\}$ such that $f'(S) = 1$ if and only if both $f(S) = 1$ and $\delta_F(S) = \emptyset$ hold. Then, f' is also pliable.*

Proof. Consider $A, B \subsetneq V$ such that $f'(A) = 1 = f'(B)$. Clearly, $f(A) = 1 = f(B)$. Moreover, for any $S \in \{A \cap B, A \cup B, A \setminus B, B \setminus A\}$, we have $\delta_F(S) = \emptyset$, by Fact 2.1. Since f is pliable, there are at least two distinct sets $S_1, S_2 \in \{A \cap B, A \cup B, A \setminus B, B \setminus A\}$ with f -value one. Then, we have $f'(S_1) = 1 = f'(S_2)$ (since $\delta_F(S_1) = \emptyset = \delta_F(S_2)$). Hence, f' is pliable. \square

Informally speaking, the next lemma states that pliable families are closed under complementation.

Lemma 2.3. *Suppose that $\mathcal{F} \subseteq 2^V$ is a pliable family. Then $\mathcal{F}^* := \mathcal{F} \cup \{(V \setminus A) \mid A \in \mathcal{F}\}$ is a pliable family.*

Proof. Consider any two sets $S_1, S_2 \in \mathcal{F}^*$. We map these two sets to two sets $A, B \in \mathcal{F}$, by taking complements if needed. Since \mathcal{F} is a pliable family, at least two of the four sets $A \cup B, A \cap B, A \setminus B, B \setminus A$ belong to \mathcal{F} . Moreover, we have $S \in \mathcal{F} \implies \bar{S} \in \mathcal{F}^*$. By examining a few cases, we can conclude that at least two of the four sets $S_1 \cup S_2, S_1 \cap S_2, S_1 \setminus S_2, S_2 \setminus S_1$ belong to \mathcal{F}^* , thus, showing that \mathcal{F}^* is a pliable family.

Next, we present the case analysis. Consider any two sets $S_1, S_2 \in \mathcal{F}^*$.

(1) $S_1 \in \mathcal{F}, S_2 \in \mathcal{F}$:

Since \mathcal{F} is a pliable family, at least two of the four sets $S_1 \cup S_2, S_1 \cap S_2, S_1 \setminus S_2, S_2 \setminus S_1$ belong to $\mathcal{F}^* \supseteq \mathcal{F}$.

(2) $\overline{S}_1 \in \mathcal{F}, \overline{S}_2 \in \mathcal{F}$:

Let $A = \overline{S}_1$ and let $B = \overline{S}_2$. Since \mathcal{F} is a pliable family, at least two of the four sets $A \cup B, A \cap B, A \setminus B, B \setminus A$ belong to \mathcal{F} . Observe the following:

$A \cap B = \overline{S}_1 \cap \overline{S}_2$; taking the complement, we have $\overline{A \cap B} = S_1 \cup S_2$; hence, $A \cap B \in \mathcal{F} \implies S_1 \cup S_2 \in \mathcal{F}^*$.

$A \cup B = \overline{S}_1 \cup \overline{S}_2$; taking the complement, we have $\overline{A \cup B} = S_1 \cap S_2$; hence, $A \cup B \in \mathcal{F} \implies S_1 \cap S_2 \in \mathcal{F}^*$.

$A \setminus B = \overline{S}_1 \setminus \overline{S}_2 = S_2 \setminus S_1$; hence, $A \setminus B \in \mathcal{F} \implies S_2 \setminus S_1 \in \mathcal{F}^*$.

$B \setminus A = \overline{S}_2 \setminus \overline{S}_1 = S_1 \setminus S_2$; hence, $B \setminus A \in \mathcal{F} \implies S_1 \setminus S_2 \in \mathcal{F}^*$.

Hence, at least two of the four sets $S_1 \cup S_2, S_1 \cap S_2, S_1 \setminus S_2, S_2 \setminus S_1$ belong to \mathcal{F}^* .

(3) $\overline{S}_1 \in \mathcal{F}, S_2 \in \mathcal{F}$:

Let $A = \overline{S}_1$ and let $B = S_2$. Since \mathcal{F} is a pliable family, at least two of the four sets $A \cup B, A \cap B, A \setminus B, B \setminus A$ belong to \mathcal{F} . Observe the following:

$A \cap B = \overline{S}_1 \cap S_2 = S_2 \setminus S_1$ hence, $A \cap B \in \mathcal{F} \implies S_2 \setminus S_1 \in \mathcal{F}^*$.

$A \cup B = \overline{S}_1 \cup S_2 = \overline{S_1 \setminus S_2}$; taking the complement, we have $\overline{A \cup B} = S_1 \setminus S_2$; hence, $A \cup B \in \mathcal{F} \implies S_1 \setminus S_2 \in \mathcal{F}^*$.

$A \setminus B = \overline{S}_1 \setminus S_2 = \overline{S_1 \cup S_2}$; taking the complement, we have $\overline{A \setminus B} = S_1 \cup S_2$; hence, $A \setminus B \in \mathcal{F} \implies S_1 \cup S_2 \in \mathcal{F}^*$.

$B \setminus A = S_2 \setminus \overline{S}_1 = S_1 \cap S_2$; hence, $B \setminus A \in \mathcal{F} \implies S_1 \cap S_2 \in \mathcal{F}^*$.

Hence, at least two of the four sets $S_1 \cup S_2, S_1 \cap S_2, S_1 \setminus S_2, S_2 \setminus S_1$ belong to \mathcal{F}^* .

□

Reducing the hitting set problem to the f -connectivity problem

The following reduction from the hitting set problem (which has a logarithmic hardness-of-approximation threshold [8]) to the f -connectivity problem shows that the (unrestricted) f -connectivity problem has a logarithmic hardness-of-approximation threshold.

In the hitting set problem, we are given a ground-set X and a family \mathcal{S} of subsets of X ; the goal is to find a smallest subset W of X such that $W \cap A$ is non-empty for each set $A \in \mathcal{S}$; in other words, the goal is to find a minimum-size cover of the sets of \mathcal{S} .

The hitting set problem reduces to an f -connectivity problem on a bipartite graph $G = (L \cup R, E)$ where we define $L := \{\ell_x : x \in X\}$, $R := \{r_x : x \in X\}$, and $E := \{e_x = \ell_x r_x : x \in X\}$; observe that E is a perfect matching of G . We define f to be the indicator function of the family $\{\{\ell_x : x \in A\} : A \in \mathcal{S}\}$.

Thus, each element $x \in X$ is modeled by an (isolated) edge $e_x = \ell_x r_x$ of the bipartite graph, and each set $A \in \mathcal{S}$ is modeled by fixing $f(\{\ell_x : x \in A\})$ to be one. Clearly, any solution $W \subseteq X$ of the hitting set problem corresponds to a solution $\{\ell_x r_x : x \in W\}$ of the f -connectivity problem of the same size.

2.2 The WGMV Primal-Dual Algorithm for Uncrossable Functions

In this section, we give a brief description of the primal-dual algorithm of Williamson et al. [5] that achieves approximation ratio 2 for any f -connectivity problem such that f is an uncrossable function.

Theorem 2.4 (Lemma 2.1 in [5]). *Let $f : 2^V \rightarrow \{0,1\}$ be an uncrossable function. Suppose we have a subroutine that for any given $F \subseteq E$, computes all minimal violated sets w.r.t. f , F . Then, in polynomial time and using a polynomial number of calls to the subroutine, we can compute a 2-approximate solution to the given instance of the f -connectivity problem.*

The algorithm and its analysis are based on the following LP relaxation of (f -IP) (stated on the left) and its dual. Define $\mathcal{S} := \{S \subseteq V : f(S) = 1\}$.

Primal LP	Dual LP
$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ \text{subject to: } & \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \in \mathcal{S} \\ & 0 \leq x_e \leq 1 \quad \forall e \in E \end{aligned}$	$\begin{aligned} \max \quad & \sum_{S \in \mathcal{S}} y_S \\ \text{subject to: } & \sum_{S \in \mathcal{S} : e \in \delta(S)} y_S \leq c_e \quad \forall e \in E \\ & y_S \geq 0 \quad \forall S \in \mathcal{S} \end{aligned}$

The algorithm starts with an infeasible primal solution $F = \emptyset$, which corresponds to $x = \chi^F = \mathbf{0} \in \{0,1\}^E$, and a feasible dual solution $y = \mathbf{0}$. At any time, we say that $S \in \mathcal{S}$ is *violated* if $\delta_F(S) = \emptyset$, i.e., the primal-covering constraint for S is not satisfied. We call inclusion-wise minimal violated sets as *active sets*. An edge $e \in E$ is said to be *tight* if $\sum_{S \in \mathcal{S} : e \in \delta(S)} y_S = c_e$, i.e., the dual-packing constraint for e is tight. Throughout the algorithm, the following conditions are maintained: (i) integrality of the primal solution; (ii) feasibility of the dual solution; (iii) y_S is never decreased for any S ; and (iv) y_S may only be increased for $S \in \mathcal{S}$ that are active.

The algorithm has two stages. In the first stage, the algorithm iteratively improves primal feasibility by including tight edges in F that are contained in the cut $\delta(S)$ of any active set S . If no such edge exists, then the algorithm uniformly increases y_S for all active sets S until a new edge becomes tight. The first stage ends when $x = \chi^F$ becomes feasible. In the second stage, called *reverse delete*, the algorithm removes redundant edges from F . Initially $F' = F$. The algorithm examines edges picked in the first stage in reverse order, and discards edges from F' as long as feasibility is maintained. Note that F' is feasible if the subroutine in the hypothesis of Theorem 2.4 does not find any (minimal) violated sets.

The analysis of the 2-approximation ratio is based on showing that a relaxed form of the complementary slackness conditions hold on “average”. Let F' and y be the primal and dual solutions returned by the algorithm. By the design of the algorithm, $\sum_{S \in \mathcal{S} : e \in \delta(S)} y_S = c_e$ holds for any edge $e \in F'$. Thus, the cost of F' can be written as $\sum_{e \in F'} \sum_{S \in \mathcal{S} : e \in \delta(S)} y_S = \sum_{S \in \mathcal{S}} y_S \cdot |\delta_{F'}(S)|$. Observe that the approximation ratio follows from showing that the algorithm always maintains the following inequality:

$$\sum_{S \in \mathcal{S}} y_S \cdot |\delta_{F'}(S)| \leq 2 \sum_{S \in \mathcal{S}} y_S. \quad (1)$$

Consider any iteration and recall that the dual variables corresponding to active sets were uniformly increased by an $\varepsilon > 0$ amount, until some edge became tight. Let \mathcal{C} denote the collection of active sets during this iteration. During this iteration, the left-hand side of (1) increases by $\varepsilon \cdot \sum_{S \in \mathcal{C}} |\delta_{F'}(S)|$ and the right-hand side increases by $2 \cdot \varepsilon \cdot |\mathcal{C}|$. Thus, (1) is maintained if one can show that the average F' -degree of active sets in every iteration is ≤ 2 , and this forms the crux of the WGMV analysis.

We refer the reader to [24] for a detailed discussion of the primal-dual method for network design problems.

3 Extending the WGMV Primal-Dual Method to Pliable functions with property (γ)

In this section, we prove our main result, Theorem 1.1: we show that the primal-dual algorithm outlined in Section 2.2 is a 16-approximation algorithm for the f -connectivity problem where f is a pliable function with property (γ) . Our analysis follows the same high-level plan as that of Williamson et al. [5] which was outlined in Section 2.2. We will show that, in any iteration of the first stage of the primal-dual algorithm,

$$\sum_{C \in \mathcal{C}} |\delta_{F'}(C)| \leq 16|\mathcal{C}|,$$

where \mathcal{C} is the collection of active sets in that iteration, and F' is the set of edges output by the algorithm at termination, i.e., after the reverse delete stage.

For the remainder of this proof we assume that the iteration, and thus \mathcal{C} , is fixed. Additionally, for notational convenience, we say that two sets A, B *overlap* if $A \setminus B, A \cap B$ and $B \setminus A$ are all non-empty. (Clearly, if A, B cross, then A, B overlap; if $A \cup B = V$, then A, B do not cross but A, B could overlap.)

We begin with a lemma that can be proved by the same arguments as in the proof of [5, Lemma 5.1].

Lemma 3.1. *For any edge $e \in \cup_{C \in \mathcal{C}} \delta_{F'}(C)$, there exists a witness set $S_e \subseteq V$ such that:*

- (i) $f(S_e) = 1$ and S_e is violated in the current iteration, and
- (ii) $\delta_{F'}(S_e) = \{e\}$.

Our proof of the following key lemma is presented in Appendix B.

Lemma 3.2. *There exists a laminar family of witness sets for the edge-set $\cup_{C \in \mathcal{C}} \delta_{F'}(C)$.*

Lemma 3.3. *The sets in \mathcal{C} are pair-wise disjoint.*

Proof. Consider two sets $C_1, C_2 \in \mathcal{C}$ such that $C_1 \cap C_2 \neq \emptyset$. Then by the definition of pliable functions, one of the sets $C_1 \cap C_2$, $C_1 \setminus C_2$, or $C_2 \setminus C_1$ is violated; thus, a proper subset of either C_1 or C_2 is violated. This is a contradiction because C_1 and C_2 are minimal violated sets and no proper subset of C_1 (respectively, C_2) is violated. \square

Let \mathcal{L} be the laminar family of witness sets (of the edge-set $\cup_{C \in \mathcal{C}} \delta_{F'}(C)$) together with the node-set V . Let \mathcal{T} be a rooted tree that represents \mathcal{L} ; for each set $S \in \mathcal{L}$, there is a node v_S in \mathcal{T} , and the node v_V is taken to be the root of \mathcal{T} . Thus, \mathcal{T} has an edge $v_Q v_S$ such that $Q \supsetneq S$ iff Q is the smallest set of \mathcal{L} that properly contains the set S of \mathcal{L} . Let ψ be a mapping from \mathcal{C} to \mathcal{L} that maps each active set C to the smallest set $S \in \mathcal{L}$ that contains it. If a node v_S of \mathcal{T} has some active set mapped to its

associated set S , then we call v_S *active* and we assign the color red to v_S . Moreover, we assign the color green to each of the non-active nodes of \mathcal{T} that are incident to three or more edges of \mathcal{T} ; thus, node v_S of \mathcal{T} is green iff $\deg_{\mathcal{T}}(v_S) \geq 3$ and v_S is not active. Finally, we assign the color black to each of the remaining nodes of \mathcal{T} ; thus, node v_S of \mathcal{T} is black iff $\deg_{\mathcal{T}}(v_S) \leq 2$ and v_S is not active.

Let the number of red, green and black nodes of \mathcal{T} be denoted by n_R, n_G and n_B , respectively. Clearly, $n_R + n_G + n_B = |\mathcal{T}| = 1 + |\cup_{C \in \mathcal{C}} \delta_{F'}(C)|$. Let n_L denote the number of leaf nodes of \mathcal{T} .

Lemma 3.4. *The following statements hold:*

- (i) *Each leaf node of \mathcal{T} is red.*
- (ii) *We have $n_G \leq n_L \leq n_R \leq |\mathcal{C}|$.*

Proof. The first statement follows by repeating the argument in [5, Lemma 5.3].

We have $n_G \leq n_L$ because the number of leaves in any tree is at least the number of nodes that are incident to three or more edges of the tree. Moreover, by (i), we have $n_L \leq n_R$. Every red node of \mathcal{T} is associated with a set in \mathcal{C} by the mapping ψ , hence, $n_R \leq |\mathcal{C}|$. \square

Observe that each black node of \mathcal{T} is incident to two edges of \mathcal{T} ; thus, every black non-root node of \mathcal{T} has a unique child.

Let us sketch our plan for proving Theorem 1.1. Clearly, the theorem would follow from the inequality $\sum_{C \in \mathcal{C}} |\delta_{F'}(C)| \leq 16 |\mathcal{C}|$; thus, we need to prove an upper bound of $16 |\mathcal{C}|$ on the number of “incidences” between the edges of F' and the cuts $\delta(C)$ of the active sets $C \in \mathcal{C}$. We start by assigning a token to \mathcal{T} corresponding to each “incidence”. In more detail, for each edge $e \in F'$ and cut $\delta(C)$ such that $C \in \mathcal{C}$ and $e \in \delta(C)$ we assign one token to the node v_{S_e} of \mathcal{T} that represents the witness set S_e of the edge e . Thus, the total number of tokens assigned to \mathcal{T} is $\sum_{C \in \mathcal{C}} |\delta_{F'}(C)|$; moreover, after the initial assignment, it can be seen that each node of \mathcal{T} has ≤ 2 tokens (see Lemma 3.5 below). Then we redistribute the tokens according to a simple rule such that (after redistributing) each of the red/green nodes has ≤ 8 tokens and each of the black nodes has no tokens. Lemma 3.6 (below) proves this key claim by applying property (γ) . The key claim implies that the total number of tokens assigned to \mathcal{T} is $\leq 8n_R + 8n_G \leq 16n_R \leq 16|\mathcal{C}|$ (by Lemma 3.4). This concludes our sketch.

We apply the following two-phase scheme to assign tokens to the nodes of \mathcal{T} .

- In the first phase, for $C \in \mathcal{C}$ and $e \in \delta_{F'}(C)$, we assign a new token to the node v_{S_e} corresponding to the witness set S_e for the edge e . At the end of the first phase, observe that the root v_V of \mathcal{T} has no tokens (since the set V cannot be a witness set).
- In the second phase, we apply a root-to-leaves scan of \mathcal{T} (starting from the root v_V). Whenever we scan a black node, then we move all the tokens at that node to its unique child node. (There are no changes to the token distribution when we scan a red node or a green node.)

Lemma 3.5. *At the end of the first phase, each node of \mathcal{T} has ≤ 2 tokens.*

Proof. Consider a non-root node v_{S_e} of \mathcal{T} . This node corresponds to a witness set $S_e \in \mathcal{L}$ and e is the unique edge of F' in $\delta(S_e)$. The edge e is in ≤ 2 of the cuts $\delta(C), C \in \mathcal{C}$, because the active sets are pairwise disjoint (in other words, the number of “incidences” for e is ≤ 2). No other edge of F' can assign tokens to v_{S_e} during the first phase. \square

Lemma 3.6. *The following statements hold:*

- (i) Consider any path of $\mathcal{T} \setminus \{v_V\}$ with four nodes $v_{S_4}, v_{S_3}, v_{S_2}, v_{S_1}$ such that $S_1 \subsetneq S_2 \subsetneq S_3 \subsetneq S_4$. At least one of the four nodes is red or green (i.e., non-black).
- (ii) Hence, after token redistribution, each red or green node of \mathcal{T} has ≤ 8 tokens and each black node of \mathcal{T} has zero tokens.

Proof. For the sake of contradiction, assume that there exists a path of $\mathcal{T} \setminus \{v_V\}$ with four black nodes $v_{S_4}, v_{S_3}, v_{S_2}, v_{S_1}$ such that $S_1 \subsetneq S_2 \subsetneq S_3 \subsetneq S_4$; clearly, S_1, S_2, S_3, S_4 are witness sets in \mathcal{L} . For $i \in \{1, 2, 3, 4\}$, let S_i be the witness set of edge $e_i = \{a_i, b_i\} \in F'$; note that e_i has exactly one end-node in S_i , call it a_i . Clearly, for $i \in \{1, 2, 3\}$, both nodes a_i, b_i are in S_{i+1} (since e_{i+1} is the unique edge of F' in $\delta(S_{i+1})$).

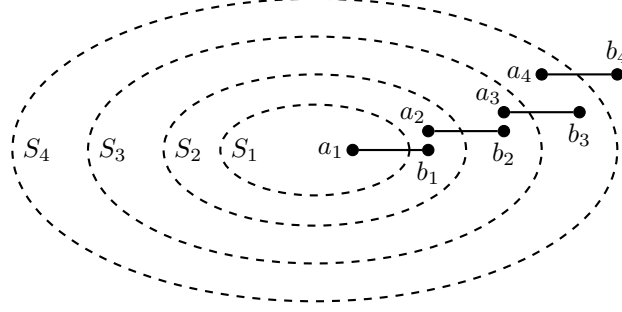


Fig. 2 Illustration of the witness sets $S_1 \subsetneq S_2 \subsetneq S_3 \subsetneq S_4$ and the edges $a_i b_i$ ($i = 1, \dots, 4$), in the proof of Lemma 3.6.

Let $C \in \mathcal{C}$ be an active set such that $e_1 \in \delta(C)$.

Claim 3.7. C is not a subset of S_1 .

For the sake of contradiction, suppose that C is a subset of S_1 . Since e_1 has (exactly) one end-node in C and $b_1 \notin S_1$, we have $a_1 \in C$. Let W be the smallest set in \mathcal{L} that contains C . Then $W \subseteq S_1$, and, possibly, $W = S_1$. Thus, we have $a_1 \in W$ and $b_1 \notin W$, hence, $e_1 \in \delta(W)$. Then we must have $W = S_1$ (since e_1 is in exactly one of the cuts $\delta(S), S \in \mathcal{L}$). Then the mapping ψ from \mathcal{C} to \mathcal{L} maps C to $W = S_1$, hence, v_{S_1} is colored red. This is a contradiction.

Claim 3.8. C crosses each of the sets S_2, S_3, S_4 .

First, observe that e_1 has (exactly) one end-node in C and has both end-nodes in S_2 . Hence, both $S_2 \cap C$ and $S_2 \setminus C$ are non-empty. Next, using Claim 3.7, we can prove that C is not a subset of S_2 . (Otherwise, S_2 would be the smallest set in \mathcal{L} that contains C , hence, v_{S_2} would be colored red.) Repeating the same argument, we can prove that C is not a subset of S_3 , and, moreover, C is not a subset of S_4 . Finally, note that $V \setminus (C \cup S_4)$ is non-empty. (Otherwise, at least one of $C \setminus S_4$ or $C \cap S_4$ would be violated, since f is a pliable function, and that would contradict the fact that C is a minimal violated set.) Observe that S_2 crosses C because all four sets $S_2 \cap C, S_2 \setminus C, C \setminus S_2, V \setminus (S_2 \cup C)$ are non-empty (in more detail, we have $|\{a_1, b_1\} \cap (S_2 \cap C)| = 1, |\{a_1, b_1\} \cap (S_2 \setminus C)| = 1, C \not\subseteq S_2 \implies C \setminus S_2 \neq \emptyset, V \setminus (C \cup S_2) \supseteq V \setminus (C \cup S_4) \neq \emptyset$). Similarly, it can be seen that S_3 crosses C , and S_4 crosses C .

Claim 3.9. Either $S_3 \setminus (C \cup S_2)$ is non-empty or $S_4 \setminus (C \cup S_3)$ is non-empty.

For the sake of contradiction, suppose that both sets $S_3 \setminus (C \cup S_2)$, $S_4 \setminus (C \cup S_3)$ are empty. Then $C \supseteq S_4 \setminus S_3$ and $C \supseteq S_3 \setminus S_2$. Consequently, both end-nodes of e_3 are in C (since $a_3 \in S_3 \setminus S_2$ and $b_3 \in S_4 \setminus S_3$). This leads to a contradiction, since $e_3 \in F'$ is incident to an active set in \mathcal{C} , call it C_3 (i.e., $e_3 \in \delta(C_3)$), hence, one of the end-nodes of e_3 is in both C and C_3 , whereas the active sets are pairwise disjoint.

To conclude the proof of the lemma, suppose that $S_4 \setminus (C \cup S_3)$ is non-empty (by Claim 3.9); the other case, namely, $S_3 \setminus (C \cup S_2) \neq \emptyset$, can be handled by the same arguments. Then, by property (γ) , $S_4 \setminus (C \cup S_3)$ is a violated set, therefore, it contains a minimal violated set, call it \tilde{C} . Clearly, the mapping ψ from \mathcal{C} to \mathcal{L} maps the active set \tilde{C} to a witness set $S_{\tilde{C}}$ which is the smallest set in \mathcal{L} that contains \tilde{C} . Either $S_{\tilde{C}} = S_4$ or else $S_{\tilde{C}}$ is a subset of $S_4 \setminus S_3$. Both cases give contradictions; in the first case, S_4 is colored red, and in the second case, v_{S_4} has ≥ 2 children in \mathcal{T} so that S_4 is colored either green or red. Thus, we have proved the first part of the lemma.

The second part of the lemma follows by Lemma 3.4 and the sketch given below Lemma 3.4. In more detail, at the start of the second phase, each node of \mathcal{T} has ≤ 2 tokens, by Lemma 3.5. In the second phase, we redistribute the tokens such that each (non-root) black node ends up with no tokens, and each red/green node v_S receives ≤ 6 redistributed tokens because there are ≤ 3 black ancestor nodes of v_S that could send their tokens to v_S (by the first part of the lemma). Hence, each non-root non-black node has ≤ 8 tokens, after token redistribution. \square

Theorem 1.1 follows from Lemmas 3.1–3.6.

4 $O(1)$ -Approximation Algorithm for Augmenting Small Cuts

In this section, we give a 16-approximation algorithm for the AugSmallCuts problem, thereby proving Theorem 1.2. Our algorithm for AugSmallCuts is based on a reduction to an instance of the f -connectivity problem on the graph $H = (V, L)$ for a pliable function f with property (γ) .

Recall the AugSmallCuts problem: we are given an undirected graph $G = (V, E)$ with edge-capacities $u \in \mathbb{Q}_{\geq 0}^E$, a set of links $L \subseteq \binom{V}{2}$ with costs $c \in \mathbb{Q}_{\geq 0}^L$, and a threshold $\tilde{\lambda} \in \mathbb{Q}_{\geq 0}$. A subset $F \subseteq L$ of links is said to *augment* 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 augments all non-empty $S \subsetneq V$ with $u(\delta_E(S)) < \tilde{\lambda}$.

Proof of Theorem 1.2. Define $f : 2^V \rightarrow \{0, 1\}$ such that $f(S) = 1$ if and only if $S \notin \{\emptyset, V\}$ and $u(\delta_E(S)) < \tilde{\lambda}$. We apply Theorem 1.1 for the f -connectivity problem on the graph $H = (V, L)$ with edge-costs $c \in \mathbb{Q}_{\geq 0}^L$ to obtain a 16-approximate solution $F \subseteq L$. By our choice of f , there is a one-to-one cost-preserving correspondence between feasible augmentations for AugSmallCuts and feasible solutions to the f -connectivity problem. Thus, it remains to argue that the assumptions of Theorem 1.1 hold.

First, we show that f is pliable. Note that f is symmetric and $f(V) = 0$. Consider sets $A, B \subseteq V$ with $f(A) = f(B) = 1$. By submodularity and symmetry of cuts in undirected graphs, we have: $\max\{u(\delta(A \cup B)) + u(\delta(A \cap B)), u(\delta(A \setminus B)) + u(\delta(B \setminus A))\} \leq u(\delta(A)) + u(\delta(B))$. Since the right hand side is strictly less than $2\tilde{\lambda}$, we have $f(A \cap B) + f(A \cup B) \geq 1$ and $f(A \setminus B) + f(B \setminus A) \geq 1$, hence, f is pliable.

Second, we argue that f satisfies property (γ) . Fix some edge-set $F \subseteq L$, and define $f' : 2^V \rightarrow \{0, 1\}$ such that $f'(S) = 1$ if and only if $f(S) = 1$ and $\delta_F(S) = \emptyset$. By Fact 2.2, f' is also pliable. Consider sets $C, S_1, S_2 \subseteq V$, $S_1 \subsetneq S_2$, that are violated w.r.t. f , F , i.e., $f'(C) = f'(S_1) = f'(S_2) = 1$. Further, suppose that C is minimally violated, and C crosses both S_1 and S_2 . Suppose that $S_2 \setminus (S_1 \cup C)$ is

non-empty (the other case is trivial). To show that $S_2 \setminus (S_1 \cup C)$ is violated w.r.t. f, F , we have to show that (i) $\delta_F(S_2 \setminus (S_1 \cup C))$ is empty and (ii) $u(\delta_E(S_2 \setminus (S_1 \cup C))) < \tilde{\lambda}$. Observe that S_2 crosses $(S_1 \cup C)$. To show (i), we apply Fact 2.1 twice; first, we show that $\delta_F(S_1 \cup C)$ is empty (since $\delta_F(C), \delta_F(S_1)$ are empty), and then we show that $\delta_F(S_2 \setminus (S_1 \cup C))$ is empty (since $\delta_F(S_2)$ is empty). To show (ii), observe that the multiset $\delta_E(S_2 \setminus (S_1 \cup C)) \cup \delta_E(C \setminus S_2)$ is a subset of the multiset $\delta_E(S_2) \cup \delta_E(C \cup S_1)$, and, for each edge, its multiplicity in $\delta_E(S_2 \setminus (S_1 \cup C)) \cup \delta_E(C \setminus S_2)$ is less than or equal to its multiplicity in $\delta_E(S_2) \cup \delta_E(C \cup S_1)$. (Note that for disjoint sets $A_1, A_2, A_3 \subseteq V$, the multiset $\delta(A_1) \cup \delta(A_2)$ is a subset of the multiset $\delta(A_1 \cup A_3) \cup \delta(A_2 \cup A_3)$.) Moreover, we claim that $u(\delta_E(C \cup S_1)) < \tilde{\lambda}$ and $u(\delta_E(C \setminus S_2)) \geq \tilde{\lambda}$. The two claims immediately imply (ii) (since $u(\delta_E(S_2)) < \tilde{\lambda}$).

Next, we prove the two claims. Note that the sets $C \cap S_1, C \setminus S_1, S_1 \setminus C, V \setminus (C \cup S_1)$ are non-empty, and note that $f'(C \cap S_1) = 0 = f'(C \setminus S_1)$ since C is a minimal violated set. Since f' is pliable and $f'(C) = 1 = f'(S_1)$, we have $f'(C \cup S_1) = 1$. By Fact 2.1, $\delta_F(C \cup S_1) = \emptyset$, hence, $f(C \cup S_1) = 1$; equivalently, $u(\delta_E(C \cup S_1)) < \tilde{\lambda}$. Since C is a minimal violated set, $f'(C \setminus S_2) = 0$. Moreover, $\delta_F(C \setminus S_2) = \emptyset$, by Fact 2.1. Hence, $f(C \setminus S_2) = 0$; equivalently, $u(\delta_E(C \setminus S_2)) \geq \tilde{\lambda}$.

Last, we describe a polynomial-time subroutine that for any $F \subseteq L$ gives the collection of all minimal violated sets w.r.t. f, F . Assign a capacity of $\tilde{\lambda}$ to all edges in F , and consider the graph $G' = (V, E')$ where $E' := E \cup F$. Then, the family of minimal violated sets is given by $\{\emptyset \subsetneq A \subsetneq V : u(\delta_{E'}(S)) < \tilde{\lambda}, u(\delta_{E'}(A)) \geq \tilde{\lambda} \forall \emptyset \subsetneq A \subsetneq S\}$. We use the notion of solid sets to find all such minimally violated sets; see Naor, Gusfield, and Martel [34] and see Frank's book [35]. A solid set of an undirected graph $H = (V, E'')$ with capacities $w \in \mathbb{R}_{\geq 0}^{E''}$ on its edges is a non-empty node-set $Z \subsetneq V$ such that $w(\delta_{E''}(X)) > w(\delta_{E''}(Z))$ for all non-empty $X \subsetneq Z$. Note that the family of minimal violated sets of interest to us is a sub-family of the family of solid sets of G' . The family of all solid sets of a graph can be listed in polynomial time, see [34] and [35, Chapter 7.3]. Hence, we can find all minimal violated sets w.r.t. f, F in polynomial time, by examining the list of solid sets to check (1) whether there is a solid set S that is violated, and (2) whether every proper subset of S that is a solid set is *not* violated. This completes the proof of the theorem. \square

5 $O(k/u_{\min})$ -Approximation Algorithm for the Capacitated k -Edge-Connected Subgraph Problem

In this section, we give a $16 \cdot \lceil k/u_{\min} \rceil$ -approximation algorithm for the Cap- k -ECSS problem, thereby proving Theorem 1.3. Our algorithm is based on repeated applications of Theorem 1.2.

Recall the capacitated k -edge-connected subgraph problem (Cap- k -ECSS): we are given an undirected graph $G = (V, E)$ with edge costs $c \in \mathbb{Q}_{\geq 0}^E$ and edge capacities $u \in \mathbb{Z}_{\geq 0}^E$. The goal is to find a minimum-cost subset of the edges $F \subseteq E$ such that the capacity across any cut in (V, F) is at least k , i.e., $u(\delta_F(S)) \geq k$ for all non-empty sets $S \subsetneq V$.

Proof of Theorem 1.3. The algorithm is as follows: Initialize $F := \emptyset$. While the minimum capacity of a cut $\delta(S)$, $\emptyset \neq S \subsetneq V$, in (V, F) is less than k , run the approximation algorithm from Theorem 1.2 with input $G = (V, F)$ and $L = E \setminus F$, to augment all cuts $\delta(S)$, $\emptyset \neq S \subsetneq V$, with $u(\delta(S)) < k$ and obtain a valid augmentation $F' \subseteq L$. Update F by adding F' , that is, $F := F \cup F'$. On exiting the while loop, output the set of edges F .

At any step of the algorithm, let λ denote the minimum capacity of a cut in (V, F) , i.e., $\lambda := \min\{u(\delta_F(S)) : \emptyset \subsetneq S \subsetneq V\}$.

The above algorithm outputs a feasible solution since, upon exiting the while loop, λ is at least k . Let $F^* \subseteq E$ be an optimal solution to the Cap- k -ECSS instance. Notice that $F^* \setminus F$ is a feasible choice for F' during any iteration of the while loop. Hence, by Theorem 1.2, $c(F') \leq 16 \cdot c(F^*)$. We claim that the above algorithm requires at most $\lceil \frac{k}{u_{\min}} \rceil$ iterations of the while loop. This holds because each iteration of the while loop (except possibly the last iteration) raises λ by at least u_{\min} . (At the start of the last iteration, $k - \lambda$ could be less than u_{\min} , and, at the end of the last iteration, λ could be equal to k). Hence, at the end of the algorithm, $c(F) \leq 16 \cdot \lceil \frac{k}{u_{\min}} \rceil c(F^*)$. This completes the proof. \square

Our new result (Theorem 1.2) is critical for the bound of $\lceil \frac{k}{u_{\min}} \rceil$ on the number of iterations of this algorithm. Earlier methods only allowed augmentations of minimum cuts, so such methods may require as many as $\Omega(k)$ iterations. (In more detail, the earlier methods would augment the cuts of (V, F) of capacity λ but would not augment the cuts of capacity $\geq \lambda + 1$; thus, cuts of capacity $\lambda + 1$ could survive the augmentation step.)

6 $O(1)$ -Approximation Algorithm for $(p, 2)$ -FGC

In this section, we present a 20-approximation algorithm for $(p, 2)$ -FGC, by applying our results from Section 3.

Recall (from Section 1) that the algorithmic goal in $(p, 2)$ -FGC is to find a minimum-cost edge-set F such that for any pair of unsafe edges $e, f \in F \cap \mathcal{U}$, the subgraph $(V, F \setminus \{e, f\})$ is p -edge connected.

Our algorithm works in two stages. First, we compute a feasible edge-set F_1 for $(p, 1)$ -FGC on the same input graph, by applying the 4-approximation algorithm of [13]. We then augment the subgraph (V, F_1) using additional edges. Since F_1 is a feasible edge-set for $(p, 1)$ -FGC, any cut $\delta(S)$, $\emptyset \subsetneq S \subsetneq V$, in the subgraph (V, F_1) either (i) has at least p safe edges or (ii) has at least $p + 1$ edges (see below for a detailed argument). Thus the cuts that need to be augmented have exactly $p + 1$ edges and contain at least two unsafe edges. Augmenting all such cuts by at least one (safe or unsafe) edge will ensure that we have a feasible solution to $(p, 2)$ -FGC.

The following example shows that when p is odd, then the function f in the f -connectivity problem associated with $(p, 2)$ -FGC may *not* be an uncrossable function.

Example 6.1. *We construct the graph G by starting with a 4-cycle v_1, v_2, v_3, v_4, v_1 and then replacing each edge of the 4-cycle by a pair of parallel edges; thus, we have a 4-regular graph with 8 edges; we designate the following four edges as unsafe (and the other four edges are safe): both copies of edge $\{v_1, v_4\}$, one copy of edge $\{v_1, v_2\}$, and one copy of edge $\{v_3, v_4\}$. Clearly, G is a feasible instance of $(3, 1)$ -FGC. On the other hand, G is infeasible for $(3, 2)$ -FGC, and the sets $\{v_1, v_2\}$ and $\{v_2, v_3\}$ are violated. Note that the indicator function f associated with the violated node-sets is not uncrossable (observe that the sets $\{v_2\}$ and $\{v_3\}$ are not violated). Moreover, observe that the minimal violated set $C = \{v_2, v_3\}$ crosses the violated set $S = \{v_1, v_2\}$.*

Proof of Theorem 1.4. In the following, we use F to denote the set of edges picked by the algorithm at any step of the execution; we mention that our correctness arguments are valid despite this ambiguous notation; moreover, we use $\delta(S)$ rather than $\delta_F(S)$ to refer to a cut of the subgraph (V, F) , where $\emptyset \neq S \subseteq V$.

Since F is a feasible edge-set for $(p, 1)$ -FGC, any cut $\delta(S)$ (where $\emptyset \neq S \subseteq V$) either (i) has at least p safe edges or (ii) has at least $p + 1$ edges. Consider a node-set S that violates the requirements of the $(p, 2)$ -FGC problem. We have $\emptyset \neq S \subsetneq V$ and there exist two unsafe edges $e, f \in \delta(S)$ such that

$|\delta(S) \setminus \{e, f\}| \leq p - 1$. Since F is feasible for $(p, 1)$ -FGC, we have $|\delta(S) \setminus \{e\}| \geq p$ and $|\delta(S) \setminus \{f\}| \geq p$. Thus, $|\delta(S)| = p + 1$. In other words, the node-sets S that need to be augmented have exactly $p + 1$ edges in $\delta(S)$, at least two of which are unsafe edges. Let $f : 2^V \rightarrow \{0, 1\}$ be the indicator function of these violated sets. Observe that f is symmetric, that is, $f(S) = f(V \setminus S)$ for any $S \subseteq V$; this additional property of f is useful for our arguments. We claim that f is a pliable function that satisfies property (γ) , hence, we obtain an $O(1)$ -approximation algorithm for $(p, 2)$ -FGC, via the primal-dual method and Theorem 1.1.

Our proof of the following lemma is presented in [29, Section 5] as well as in Appendix C

Lemma 6.1. *f is a pliable function that satisfies property (γ) . Moreover, for even p , f is an uncrossable function.*

Lastly, we show that there is a polynomial-time subroutine for computing the minimal violated sets. Consider the graph (V, F) . Note that size of a minimum cut of (V, F) is at least p since F is a feasible edge-set for $(p, 1)$ -FGC. The violated sets are subsets $S \subseteq V$ such that $\delta(S)$ contains exactly $p + 1$ edges, at least two of which are unsafe edges. Clearly, all the violated sets are contained in the family of sets S such that $\delta(S)$ is a 2-approximate min-cut of (V, F) ; in other words, $\{S \subseteq V : p \leq |\delta(S)| \leq 2p\}$ contains all the violated sets. It is well known that the family of 2-approximate min-cuts in a graph can be listed in polynomial time, see [36, 37]. Hence, we can find all violated sets and all minimally violated sets in polynomial time.

Thus, we have a 20-approximation algorithm for $(p, 2)$ -FGC via the primal-dual algorithm of [5] based on our results in Section 3. In more detail, we first find a solution to a $(p, 1)$ -FGC instance of cost $\leq 4\text{OPT}$, and then, by applying Lemma 6.1 and Theorem 1.1, we find a solution to a $(p, 2)$ -FGC instance of cost $\leq 16\text{OPT}$. Moreover, for even p , the approximation ratio is 6 ($= 4 + 2$) since the $(p, 2)$ -FGC instance corresponds to an f -connectivity problem such that f is an uncrossable function (by Lemma 6.1), hence, we find a solution of cost $\leq 2\text{OPT}$ (by Theorem 2.4). This completes the proof of Theorem 1.4. \square

Funding

The first author is partially supported by Office of Naval Research (ONR) Grant N00014-21-1-2575. The second author is supported in part by NSERC, RGPIN-2019-04197. The fourth author received funding from the following sources: NSERC grant 327620-09 and an NSERC DAS Award, European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement no. ScaleOpt-757481), and Dutch Research Council NWO Vidi Grant 016.Vidi.189.087.

Conflicts of interest / Competing interests

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Acknowledgments

We thank the anonymous reviewers and the ICALP PC for their comments. We are grateful to Cedric Koh and Madison Van Dyk for reading a preliminary version, and for their detailed comments and feedback.

A preliminary version of this paper appeared in the Proceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP 2023), Ed. K.Etessami, U.Feige, and G.Puppis <https://drops.dagstuhl.de/opus/volltexte/2023/18067/> (LIPIcs, Volume 261, Article No. 15, pp. 15:1–15:19), [38].

References

- [1] Kuhn, H.W.: The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly* **2**(1-2), 83–97 (1955) <https://doi.org/10.1002/nav.3800020109>
- [2] Dantzig, G.B., Jr., L.R.F., Fulkerson, D.R.: A Primal-Dual Algorithm for Linear Programs. In: *Linear Inequalities and Related Systems*. *Annals of Mathematics Studies*, vol. 38, pp. 171–182. Princeton University Press, Princeton, US (1957). <https://doi.org/10.1515/9781400881987-008>
- [3] Agrawal, A., Klein, P., Ravi, R.: When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks. *SIAM Journal on Computing* **24**(3), 440–456 (1995) <https://doi.org/10.1137/S0097539792236237>
- [4] Goemans, M.X., Williamson, D.P.: A General Approximation Technique for Constrained Forest Problems. *SIAM Journal on Computing* **24**(2), 296–317 (1995) <https://doi.org/10.1137/S0097539793242618>
- [5] Williamson, D.P., Goemans, M.X., Mihail, M., Vazirani, V.V.: A Primal-Dual Approximation Algorithm for Generalized Steiner Network Problems. *Combinatorica* **15**(3), 435–454 (1995) <https://doi.org/10.1007/BF01299747>
- [6] Lau, L.C., Ravi, R., Singh, M.: *Iterative Methods in Combinatorial Optimization*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, UK (2011). <https://doi.org/10.1017/CBO9780511977152>
- [7] Vazirani, V.V.: *Approximation Algorithms*. Springer, Berlin Heidelberg New York (2003). <https://doi.org/10.1007/978-3-662-04565-7>
- [8] Williamson, D.P., Shmoys, D.B.: *The Design of Approximation Algorithms*. Cambridge University Press, Cambridge, UK (2011)
- [9] Jain, K.: A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. *Combinatorica* **21**(1), 39–60 (2001) <https://doi.org/10.1007/s004930170004>
- [10] Gabow, H.N., Goemans, M.X., Tardos, É., Williamson, D.P.: Approximating the Smallest k -Edge Connected Spanning Subgraph by LP-Rounding. *Networks* **53**(4), 345–357 (2009) <https://doi.org/10.1002/net.20289>
- [11] Gabow, H.N., Gallagher, S.: Iterated Rounding Algorithms for the Smallest k -Edge Connected Spanning Subgraph. *SIAM Journal on Computing* **41**(1), 61–103 (2012) <https://doi.org/10.1137/080732572>
- [12] Fleischer, L., Jain, K., Williamson, D.P.: Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences* **72**(5), 838–867 (2006) <https://doi.org/10.1016/j.jcss.2005.05.006>

- [13] Boyd, S.C., Cheriyan, J., Haddadan, A., Ibrahimpur, S.: Approximation algorithms for flexible graph connectivity. *Mathematical Programming* (2023) <https://doi.org/10.1007/s10107-023-01961-5>
- [14] Frederickson, G.N., Jájá, J.F.: Approximation Algorithms for Several Graph Augmentation Problems. *SIAM Journal on Computing* **10**(2), 270–283 (1981) <https://doi.org/10.1137/0210019>
- [15] Khuller, S., Thurimella, R.: Approximation Algorithms for Graph Augmentation. *Journal of Algorithms* **14**(2), 214–225 (1993) <https://doi.org/10.1006/jagm.1993.1010>
- [16] Benczúr, A.A., Goemans, M.X.: Deformable Polygon Representation and Near-Mincuts. In: *Building Bridges. Bolyai Society Mathematical Studies*, pp. 103–135. Springer, Berlin Heidelberg New York (2008). https://doi.org/10.1007/978-3-540-85221-6_3
- [17] Chakrabarty, D., Chekuri, C., Khanna, S., Korula, N.: Approximability of Capacitated Network Design. *Algorithmica* **72**(2), 493–514 (2015) <https://doi.org/10.1007/s00453-013-9862-4>
- [18] Khuller, S., Vishkin, U.: Biconnectivity Approximations and Graph Carvings. *Journal of the ACM* **41**(2), 214–235 (1994) <https://doi.org/10.1145/174652.174654>
- [19] Goemans, M.X., Goldberg, A.V., Plotkin, S.A., Shmoys, D.B., Tardos, É., Williamson, D.P.: Improved Approximation Algorithms for Network Design Problems. In: *Proceedings of the 5th Symposium on Discrete Algorithms*, pp. 223–232 (1994)
- [20] Adjashvili, D., Hommelsheim, F., Mühenthaler, M.: Flexible Graph Connectivity. *Mathematical Programming* **192**, 409–441 (2022) <https://doi.org/10.1007/s10107-021-01664-9>
- [21] Chekuri, C., Jain, R.: Approximation Algorithms for Network Design in Non-Uniform Fault Models. In: *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming*, vol. 261, article 36, pp. 1–20 (2023). <https://doi.org/10.4230/LIPIcs.ICALP.2023.36>
- [22] Chekuri, C., Jain, R.: Augmentation based Approximation Algorithms for Flexible Network Design. *CoRR* **abs/2209.12273** (2022) <https://doi.org/10.48550/arXiv.2209.12273>
- [23] Chekuri, C., Jain, R.: Approximating Flexible Graph Connectivity via Rácke Tree based Rounding. *CoRR* **abs/2211.08324** (2022) <https://doi.org/10.48550/arXiv.2211.08324>
- [24] Goemans, M.X., Williamson, D.P.: The Primal-Dual Method for Approximation Algorithms and Its Application to Network Design Problems. In: *Approximation Algorithms for NP-Hard Problems*, pp. 144–191. PWS Publishing Company, Boston, MA (1996). Chap. 4
- [25] Gabow, H.N., Goemans, M.X., Williamson, D.P.: An efficient approximation algorithm for the survivable network design problem. *Mathematical Programming* **82**, 13–40 (1998) <https://doi.org/10.1007/BF01585864>
- [26] Johnson, D.S., Minkoff, M., Phillips, S.: The Prize Collecting Steiner Tree Problem: Theory and Practice. In: Shmoys, D.B. (ed.) *Proceedings of the 11th Symposium on Discrete Algorithms*, pp. 760–769 (2000). <https://dl.acm.org/doi/10.5555/338219.338637>
- [27] Mihail, M., Shallcross, D., Dean, N., Mostrel, M.: A Commercial Application of Survivable Network Design: ITP/INPLANS CCS Network Topology Analyzer. In: *Proceedings of the 7th Symposium on Discrete Algorithms*, pp. 279–287 (1996). <https://dl.acm.org/doi/10.5555/313852.314074>
- [28] Williamson, D.P., Goemans, M.X.: Computational Experience with an Approximation Algorithm on Large-Scale Euclidean Matching Instances. *INFORMS Journal on Computing* **8**(1), 29–40 (1996) <https://doi.org/10.1287/ijoc.8.1.29>
- [29] Bansal, I., Cheriyan, J., Grout, L., Ibrahimpur, S.: Improved Approximation Algorithms by Generalizing the Primal-Dual Method Beyond Uncrossable Functions. *CoRR* **abs/2209.11209v2** (2022)

- [30] Bansal, I.: A constant factor approximation for the $(p, 3)$ -flexible graph connectivity problem. CoRR **abs/2308.15714** (2023) <https://doi.org/10.48550/ARXIV.2308.15714>
- [31] Nutov, Z.: Improved approximation algorithms for some capacitated k edge connectivity problems. CoRR **abs/2307.01650** (2023) <https://doi.org/10.48550/ARXIV.2307.01650>
- [32] Diestel, R.: Graph Theory. Graduate Texts in Mathematics. Springer, Berlin Heidelberg New York (2017). <https://doi.org/10.1007/978-3-662-53622-3>
- [33] Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Algorithms and Combinatorics, vol. 24. Springer, Berlin Heidelberg New York (2003)
- [34] Naor, D., Gusfield, D., Martel, C.: A Fast Algorithm for Optimally Increasing the Edge Connectivity. SIAM Journal on Computing **26**(4), 1139–1165 (1997) <https://doi.org/10.1137/S0097539792234226>
- [35] Frank, A.: Connections in Combinatorial Optimization. Oxford Lecture Series in Mathematics and Its Applications, vol. 38. Oxford University Press, Oxford New York (2011)
- [36] Karger, D.R., Stein, C.: A New Approach to the Minimum Cut Problem. Journal of the ACM **43**(4), 601–640 (1996) <https://doi.org/10.1145/234533.234534>
- [37] Nagamochi, H., Nishimura, K., Ibaraki, T.: Computing All Small Cuts in an Undirected Network. SIAM Journal on Discrete Mathematics **10**(3), 469–481 (1997) <https://doi.org/10.1137/S0895480194271323>
- [38] Bansal, I., Cheriyan, J., Grout, L., Ibrahimpur, S.: Improved Approximation Algorithms by Generalizing the Primal-Dual Method Beyond Uncrossable Functions. In: Proceedings of the 50th International Colloquium on Automata, Languages, and Programming, vol. 261, article 15, pp. 1–19 (2023). <https://doi.org/10.4230/LIPIcs.ICALP.2023.15>

A Pliable families: an example

In Section 3, we showed an approximation ratio of $O(1)$ for the WGMV primal-dual method applied to any f -connectivity problem where f is a pliable function satisfying property (γ) . A natural question arises: Is property (γ) essential for this result? In other words, does the WGMV primal-dual method for the f -connectivity problem where f is a pliable function achieve an approximation ratio of $O(1)$? The next construction shows that the answer is **no**; thus, property (γ) is essential for our results in Section 3.

Recall that a family of sets $\mathcal{F} \subseteq 2^V$ is called a pliable family if $A, B \in \mathcal{F}$ implies that at least two of the four sets $A \cup B, A \cap B, A \setminus B, B \setminus A$ also belong to \mathcal{F} .

Now, we describe the construction: We have node-sets C_{ij} for $i = 1, 2, \dots, k$ and $j = 0, 1, 2, \dots, k$. All these C -sets are pair-wise disjoint. Additionally, for $j' = 1, \dots, k$, we have node-sets $T_{1j'} \subsetneq T_{2j'} \subsetneq \dots \subsetneq T_{kj'}$ and each of these are disjoint from all the C -sets. Additionally, we have at least one node v_0 lying outside the union of all these C -sets and T -sets. See Figure 3. We designate \mathcal{F}' to be the following family of node-sets that consists of two types of sets:

$$\begin{aligned}
 (I) \quad & C_i := \bigcup_{j=0}^k C_{ij} \quad \text{for } i \in [k] \\
 (II) \quad & T_{i'j'} \cup \bigcup_{(i,j) \in R'} C_{ij} \quad \text{for } i' \in [k], j' \in [k] \text{ and} \\
 & R' \subseteq R(i') \text{ where } R(i') = \{(i, j) \mid 1 \leq i < i', 0 \leq j \leq k\}
 \end{aligned}$$

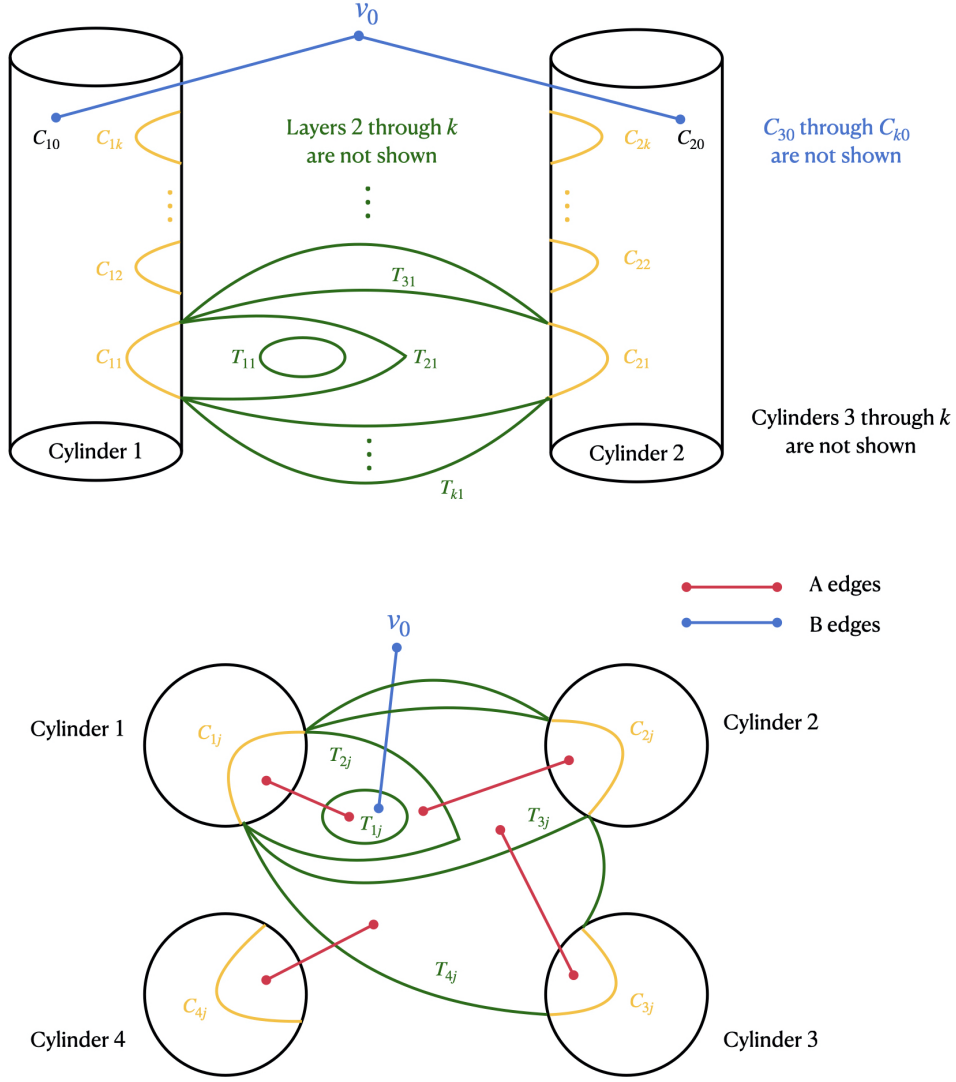


Fig. 3 Top: A bad example for the primal-dual method for augmenting a pliable family.
Bottom: Edges of type (A) and type (B) in a bad example for the primal-dual method for augmenting a pliable family.

Informally speaking, the sets C_1, C_2, \dots, C_k can be viewed as (pairwise-disjoint) “cylinders”, see Figure 3, the (first) index i is associated with one of these cylinders, and note that $C_{i0} = C_i \setminus (\cup_{j=1}^k C_{ij})$; the (second) index j is associated with a “layer” (i.e., a horizontal plane), and the sub-family $T_{1j'} \subsetneq T_{2j'} \subsetneq \dots \subsetneq T_{kj'}$ forms a nested family on layer j' , see Figure 3. For notational convenience, let $T_{0j'} = \emptyset$, $\forall j' \in [k]$. Observe that a set of type (II) is the union of one set $T_{i'j'}$ of the nested family of layer j' together with the sets of an arbitrary sub-family of each of the “cylinder families” $\{C_{i0}, C_{i1}, C_{i2}, \dots, C_{ik}\}$ with (first) index $i < i'$. We claim that the family \mathcal{F}' satisfies the condition of

Lemma 2.3. Indeed consider $A, B \in \mathcal{F}'$. If A and B are of type (I), then they are disjoint. If A is of type (I) and B is of type (II) such that $A \cap B \neq \emptyset$, then $A \cup B$ and $B \setminus A$ are sets of type (II) so both belong to \mathcal{F}' . If A and B are of type (II) such that both have the same (second) index j' , then both $A \cup B$ and $A \cap B$ are sets of type (II). On the other hand, if A and B are of type (II) such that the (second) index j' is different for A, B , then $A \setminus B$ and $B \setminus A$ are sets of type (II). Thus, by Lemma 2.3, adding in all the complements of the sets of this family gives us a pliable family \mathcal{F} .

The edges of the graph are of two types:

- (A) For $i' = 1, \dots, k$ and $j' = 1, \dots, k$, we have an edge from $T_{i'j'} \setminus T_{(i'-1)j'}$ to C_{ij} where $i = i'$ and $j = j'$.
- (B) For $j = 1, \dots, k$, we have an edge from v_0 to T_{1j} . For $i = 1, \dots, k$, we have an edge from v_0 to C_{i0} .

When the primal-dual algorithm is applied to this instance, then it picks all the edges of type (A); note that there are k^2 edges of type (A). Let us sketch the working of the primal-dual algorithm on this instance.

- Initially, the active sets are the type (I) sets C_1, \dots, C_k and the smallest T -sets T_{11}, \dots, T_{1k} . The algorithm increases the dual variables of each of these sets by $1/2$ and then picks all the type (A) edges between T_{1j} and C_{1j} for $j \in [k]$.
- In the next iteration, the (new) active sets are the type (I) sets C_2, \dots, C_k and the type (II) sets of the form $T_{2j} \cup C_{1j}$ for $j \in [k]$. The algorithm increases the dual variables of each of these sets by $1/4$ and then picks all the type (A) edges between T_{2j} and C_{2j} for $j \in [k]$.
- Similarly, in the i^{th} iteration, the active sets are the type (I) sets C_i, \dots, C_k and the type (II) sets of the form $T_{ij} \cup C_{1j} \cup \dots \cup C_{(i-1)j}$ for $j \in [k]$. The algorithm increases the dual variables of each of these sets by $1/2^i$ and then picks all the type (A) edges between T_{ij} and C_{ij} for $j \in [k]$.
- Finally, the reverse-delete step does not delete any edge, because the edges of type (A) form an inclusion-wise minimal solution.

On the other hand, all the edges of type (B) form a feasible solution of this instance, and there are $\leq 2k$ such edges. In more detail, each of the type (I) sets C_i is covered by the type (B) edge between v_0 and C_{i0} , and each of the type (II) sets containing $T_{i'j'}$ is covered by the type (B) edge between v_0 and $T_{1j'}$. Thus, the optimal solution picks $\leq 2k$ edges.

B Missing Proofs from Section 3

This section has several lemmas and proofs from Section 3 that are used to prove our main result, Theorem 1.1.

Lemma B.1. *Suppose S_1 is a witness for edge e_1 and S_2 is a witness for edge e_2 such that S_1 overlaps S_2 . Then there exist S'_1 and S'_2 satisfying the following properties:*

- (i) S'_1 is a valid witness for edge e_1 , S'_2 is a valid witness for edge e_2 , and S'_1 does not overlap S'_2 .
- (ii) $S'_1, S'_2 \in \{S_1, S_2, S_1 \cup S_2, S_1 \cap S_2, S_1 \setminus S_2, S_2 \setminus S_1\}$.
- (iii) either $S'_1 = S_1$ or $S'_2 = S_2$.

Proof. We prove the lemma via an exhaustive case analysis. Note that at least two of the four sets $S_1 \cup S_2, S_1 \cap S_2, S_1 \setminus S_2, S_2 \setminus S_1$ must be violated in the current iteration. Moreover, observe that $e_1 \in \delta_E(S_1 \setminus S_2)$ or $e_1 \in \delta_E(S_1 \cap S_2)$, and in the latter case $e_1 \in E(S_1 \cap S_2, V \setminus (S_1 \cup S_2))$ or $e_1 \in E(S_1 \cap S_2, S_2 \setminus S_1)$. We consider the following cases.

1. If $S_1 \cup S_2$ and $S_1 \cap S_2$ are violated or $S_1 \setminus S_2$ and $S_2 \setminus S_1$ are violated, then the proof of Lemma 5.2 in [5] can be applied.
2. Suppose $S_1 \cap S_2$ is violated, and one of $S_1 \setminus S_2$ or $S_2 \setminus S_1$ is violated. W.l.o.g. suppose $S_1 \cap S_2$ and $S_1 \setminus S_2$ are violated (the other case is similar). Consider where the end-nodes of the edge e_1 lie. If $e_1 \in \delta_E(S_1 \setminus S_2)$, then fix $S'_1 := S_1 \setminus S_2$ and $S'_2 := S_2$; otherwise, $e_1 \in \delta_E(S_1 \cap S_2)$, and then fix $S'_1 := S_1 \cap S_2$ and $S'_2 := S_2$.
3. Suppose $S_1 \cup S_2$ is violated, and one of $S_1 \setminus S_2$ or $S_2 \setminus S_1$ is violated. W.l.o.g. suppose $S_1 \cup S_2$ and $S_1 \setminus S_2$ are violated (the other case is similar). Consider where the end-nodes of the edges e_1 and e_2 lie. If $e_1 \in \delta_E(S_1 \setminus S_2)$, then fix $S'_1 := S_1 \setminus S_2$ and $S'_2 := S_2$; similarly, if $e_1 \in E(S_1 \cap S_2, V \setminus (S_1 \cup S_2))$, then fix $S'_1 := S_1 \cup S_2$ and $S'_2 := S_2$; finally, if $e_1 \in E(S_1 \cap S_2, S_2 \setminus S_1)$, then fix $S'_1 := S_1$, and then if $e_2 \in \delta_E(S_1 \cup S_2)$, then fix $S'_2 := S_1 \cup S_2$, otherwise, fix $S'_2 := S_1 \setminus S_2$.

This completes the proof of the lemma. \square

Lemma B.2. *Suppose a set A_1 overlaps a set A_2 and a third set A_3 does not overlap A_1 nor A_2 . Then A_3 does not overlap any of the sets $A_1 \cup A_2, A_1 \cap A_2, A_1 \setminus A_2, A_2 \setminus A_1$.*

Proof. Note that since A_3 does not overlap A_1 (or A_2), they are either disjoint or one contains the other. We consider the following cases.

1. Suppose $A_3 \cap A_1 = \emptyset$. Then $A_2 \not\subseteq A_3$ since $A_1 \cap A_2 \neq \emptyset$. If $A_3 \cap A_2 = \emptyset$, then $A_3 \subseteq V \setminus A_1 \cup A_2$ and we are done. Finally if $A_3 \subseteq A_2$, then $A_3 \subseteq A_2 \setminus A_1$ and we are done.
2. Suppose $A_1 \subseteq A_3$. Then $A_3 \cap A_2 \neq \emptyset$ since $A_1 \cap A_2 \neq \emptyset$. Also, $A_3 \not\subseteq A_2$ since $A_1 \not\subseteq A_2$. If $A_2 \subseteq A_3$, then $A_1 \cup A_2 \subseteq A_3$ and we are done.
3. Suppose $A_3 \subseteq A_1$. Then $A_2 \not\subseteq A_3$ since $A_2 \setminus A_1 \neq \emptyset$. If $A_3 \subseteq A_2$, then $A_3 \subseteq A_1 \cap A_2$ and we are done. Finally if $A_3 \cap A_2 = \emptyset$, then $A_3 \subseteq A_1 \setminus A_2$ and we are done.

\square

Lemma 3.2. *There exists a laminar family of witness sets.*

Proof. We show that any witness family can be transformed to a laminar family by repeatedly applying Lemma B.1. We prove this by induction on the size of the witness family ℓ .

Base Case: Suppose $\ell = 2$, then one application of Lemma B.1 is sufficient.

Inductive Hypothesis: If S_1, \dots, S_ℓ are witness sets for edges e_1, \dots, e_ℓ respectively with $\ell \leq k$, then, by repeatedly applying Lemma B.1, one can construct witness sets S'_1, \dots, S'_ℓ for the edges e_1, \dots, e_ℓ respectively such that S'_1, \dots, S'_ℓ is a laminar family.

Inductive Step: Consider $k+1$ witness sets S_1, \dots, S_{k+1} . By the inductive hypothesis, we can repeatedly apply Lemma B.1 to all the witness sets S_1, \dots, S_k and obtain witness sets S'_1, \dots, S'_k that form a laminar family. We now consider the following cases.

1. If S_{k+1} does not overlap some S'_i , say S'_1 , then we can apply the inductive hypothesis to the k sets $S'_2, \dots, S'_k, S_{k+1}$ and we obtain a laminar family of witness sets, none of which overlap S'_1 either (by Lemma B.2) and so we are done.
2. Suppose S_{k+1} overlaps all the sets S'_1, \dots, S'_k and for some S'_i , say S'_1 , applying Lemma B.1 to the pair S'_1, S_{k+1} gives S'_1, S'_{k+1} . Then S'_1 does not overlap any of the witness sets S'_2, \dots, S'_{k+1} , hence, applying the inductive hypothesis to these k sets gives us a laminar family of witness sets S'_2, \dots, S'_k . By Lemma B.2, S'_1 does not overlap any of the sets S'_2, \dots, S'_k and so we are done.
3. Suppose S_{k+1} overlaps all the sets S'_1, \dots, S'_k and, for every S'_i , applying Lemma B.1 to the pair S'_i, S_{k+1} gives S''_i, S_{k+1} . Then after doing this for every S'_i , we end up with the witness family

$S_1'', \dots, S_k'', S_{k+1}$ with the property that S_{k+1} does not overlap any of the other sets. Applying the inductive hypothesis to the k sets S_1'', \dots, S_k'' gives us a laminar family of witness sets S_1''', \dots, S_k''' . By Lemma B.2, S_{k+1} does not overlap any of the sets S_1''', \dots, S_k''' and so we are done. \square

C Proof of Lemma 6.1 from Section 6

This section has a proof of Lemma 6.1. This proof is the same as the proof we posted on Arxiv in 2022, [29].

Lemma 6.1. *f is a pliable function that satisfies property (γ) . Moreover, for even p , f is an uncrossable function.*

Proof. Consider two violated sets $A, B \subsetneq V$. W.l.o.g. we may assume that both A and B contain a fixed node $r \in V$. If A, B do not cross, then it is easily seen that $f(A) + f(B) = \max(f(A \cap B) + f(A \cup B), f(A \setminus B) + f(B \setminus A))$, hence, the inequality for pliable functions holds for A, B . Thus, we may assume that A, B cross. The following equations hold, see Frank's book [35, Chapter 1.2]:

$$|\delta(A)| = |\delta(B)| = p + 1 \quad (2)$$

$$|\delta(A \cup B)| + |\delta(A \cap B)| + 2|F(A \setminus B, B \setminus A)| = |\delta(A)| + |\delta(B)| \quad (3)$$

$$|\delta(A \setminus B)| + |\delta(B \setminus A)| + 2|F(A \cap B, V \setminus (A \cup B))| = |\delta(A)| + |\delta(B)| \quad (4)$$

$$|\delta(A \setminus B)| + |\delta(A \cap B)| = |\delta(A)| + 2|F(A \setminus B, A \cap B)| \quad (5)$$

Since $|\delta(S)| \geq p, \forall \emptyset \neq S \subsetneq V$, equations (2), (3) and (4) imply that $|\delta(A \cup B)|, |\delta(A \cap B)|, |\delta(A \setminus B)|, |\delta(B \setminus A)| \in \{p, p+1, p+2\}$, and, moreover, $|F(A \setminus B, B \setminus A)| \leq 1, |F(A \cap B, V \setminus (A \cup B))| \leq 1$.

Furthermore, the above four equations imply the following parity-equations (equations (6), (7) and (8) follow from equations (3), (4) and (5), respectively).

$$|\delta(A \cup B)| \equiv |\delta(A \cap B)| \pmod{2} \quad (6)$$

$$|\delta(A \setminus B)| \equiv |\delta(B \setminus A)| \pmod{2} \quad (7)$$

$$|\delta(A \cap B)| \equiv |\delta(A \setminus B)| + |\delta(A)| \equiv |\delta(B \setminus A)| + |\delta(B)| \pmod{2} \quad (8)$$

Case 1 (of proof of the lemma): Suppose that p is even. Then the parity-equations (6)–(8) imply that among the two pairs of cuts $\{\delta(A \cup B), \delta(A \cap B)\}$ and $\{\delta(A \setminus B), \delta(B \setminus A)\}$, one of the pairs consists of two $(p+1)$ -cuts, and the other pair has at least one cut of size p . Since F is a feasible solution of $(p, 1)$ -FGC, every p -cut of (V, F) consists of safe edges (that is, a p -cut cannot contain any unsafe edge). Since A and B are violated sets, each of the cuts $\delta(A)$ and $\delta(B)$ contains at least two unsafe edges.

Claim C.1. *Each cut $\delta(S)$ of the pair of $(p+1)$ -cuts $\{\delta(A \cup B), \delta(A \cap B)\}$ or $\{\delta(A \setminus B), \delta(B \setminus A)\}$ (obtained from A, B) also contains at least two unsafe edges, and so S is a violated set for each of these cuts.*

We prove this claim by a simple case analysis on the end-nodes of the unsafe edges of the cuts $\delta(A)$ and $\delta(B)$. W.l.o.g. assume that $\delta(A \cup B)$ and $\delta(A \cap B)$ are $(p+1)$ -cuts and that $\delta(A \setminus B)$ is a p -cut. The other cases are handled similarly. Clearly, $\delta(A \setminus B)$ has no unsafe edges. Since A is a violated

set, either (i) $F(A \cap B, B \setminus A)$ has two unsafe edges or (ii) $F(A \cap B, B \setminus A)$ has one unsafe edge and $F(A \cap B, V \setminus (A \cup B))$ has one unsafe edge (recall that the size of the latter edge-set is ≤ 1). Since B is a violated set, either (iii) $F(B \setminus A, V \setminus (A \cup B))$ has two unsafe edges or (iv) $F(B \setminus A, V \setminus (A \cup B))$ has one unsafe edge and $F(A \cap B, V \setminus (A \cup B))$ has one unsafe edge. Now, observe that both $A \cap B$ and $A \cup B$ are violated sets, because, by (i) or (ii), $\delta(A \cap B)$ has at least two unsafe edges, and, by (iii) or (iv), $\delta(A \cup B)$ has at least two unsafe edges. (If both $\delta(A \setminus B)$ and $\delta(B \setminus A)$ are $(p+1)$ -cuts, and there are no unsafe edges in either $\delta(A \cap B)$ or $\delta(A \cup B)$, then both $\delta(A \setminus B)$ and $\delta(B \setminus A)$ have ≥ 2 unsafe edges, so both $A \setminus B$ and $B \setminus A$ are violated sets.) This proves the claim.

Therefore, when p is even, the function f is an uncrossable function (recall that every uncrossable function is a pliable function that satisfies property (γ)).

Case 2 (of proof of the lemma): Suppose that p is odd. Then we have $|\delta(A \cup B)| \equiv |\delta(A \cap B)| \equiv |\delta(A \setminus B)| \equiv |\delta(B \setminus A)| \pmod{2}$. Hence, the above equations (2)–(5) and parity-equations (6)–(8) imply that either all four cuts $\delta(A \cup B), \delta(A \cap B), \delta(A \setminus B), \delta(B \setminus A)$ are $(p+1)$ -cuts, or at least one cut from each pair $\{\delta(A \cup B), \delta(A \cap B)\}$ and $\{\delta(A \setminus B), \delta(B \setminus A)\}$ is a p -cut.

Claim C.2. *Suppose that at least one cut from each pair $\{\delta(A \cup B), \delta(A \cap B)\}$ and $\{\delta(A \setminus B), \delta(B \setminus A)\}$ is a p -cut. Then either A is not a violated set, or B is not a violated set.*

To prove this claim, let us assume that $|\delta(A \setminus B)| = p$; similar arguments apply for the other case. The edge-set F is feasible for $(p, 1)$ -FGC, hence, all edges of $\delta(A \setminus B)$ are safe. Moreover, one of the cuts $\delta(A \cap B)$ or $\delta(A \cup B)$ has size p and consists of safe edges. If $|\delta(A \cap B)| = p$, then the cut $\delta(A)$ consists of safe edges (since $\delta(A) \subseteq \delta(A \cap B) \cup \delta(A \setminus B)$), hence, A cannot be a violated set. Otherwise, if $|\delta(A \cup B)| = p$, then the cut $\delta(B)$ consists of safe edges (since $\delta(B) = \delta(V \setminus B) \subseteq \delta(A \setminus B) \cup \delta(V \setminus (A \cup B))$), hence, B cannot be a violated set. This proves the claim.

Claim C.2 implies that all four cuts $\delta(A \cup B), \delta(A \cap B), \delta(A \setminus B), \delta(B \setminus A)$ are $(p+1)$ -cuts.

Claim C.3. *Consider two crossing violated sets $A, B \subseteq V$.*

- (i) *Then each of the four cuts $\delta(A \cup B), \delta(A \cap B), \delta(A \setminus B), \delta(B \setminus A)$ has size $(p+1)$, and we have $|F(A \setminus B, B \setminus A)| = 0 = |F(A \cap B, V \setminus (A \cup B))|$.*
- (ii) *Moreover, each of the four sets $F(A \cap B, A \setminus B), F(A \cap B, B \setminus A), F(A \setminus B, V \setminus (A \cup B)), F(B \setminus A, V \setminus (A \cup B))$ has size $(p+1)/2$.*

Part (i) of this claim follows from the above arguments (see the discussion before and after Claim C.2). Moreover, we have $|F(A \setminus B, B \setminus A)| = 0$ and $|F(A \cap B, V \setminus (A \cup B))| = 0$ by equations (3), (4). Next, consider part (ii) of the claim. Using the first part of the claim together with equation (5), we have $|F(A \cap B, A \setminus B)| = (p+1)/2$. Since $\delta(A \setminus B)$ is a $(p+1)$ -cut and it is the disjoint union of $F(A \setminus B, A \cap B)$ and $F(A \setminus B, V \setminus (A \cup B))$, we have $|F(A \setminus B, V \setminus (A \cup B))| = (p+1)/2$. Similarly, we have $|F(A \cap B, B \setminus A)| = (p+1)/2$, $|F(B \setminus A, V \setminus (A \cup B))| = (p+1)/2$. This proves the claim.

Claim C.4. (i) *At least two of the four cuts $\delta(A \cup B), \delta(A \cap B), \delta(A \setminus B), \delta(B \setminus A)$ each contain at least two unsafe edges.*

- (ii) *Moreover, if A is a minimal violated set, then $F(B \setminus A, V \setminus (A \cup B))$ has at least two unsafe edges, and each of $F(A \cap B, B \setminus A)$ and $F(A \setminus B, V \setminus (A \cup B))$ has exactly one unsafe edge.*

We prove this claim by a simple case analysis on the end-nodes of the unsafe edges of the cuts $\delta(A)$ and $\delta(B)$. Observe that $|F(A \setminus B, B \setminus A)| = 0$ and $|F(A \cap B, V \setminus (A \cup B))| = 0$, by equations (3), (4). Each edge of $\delta(A)$ is in exactly one of the sets $F(A \cap B, B \setminus A)$ or $F(A \setminus B, V \setminus (A \cup B))$; call these edge-sets ϕ_1, ϕ_2 for notational convenience. Similarly, each edge of $\delta(B)$ is in exactly one of the sets

$F(A \cap B, A \setminus B)$ or $F(B \setminus A, V \setminus (A \cup B))$; call these edge-sets ϕ_3, ϕ_4 for notational convenience. If two of the unsafe edges of $\delta(A)$ are in the same set (i.e., if ϕ_1 or ϕ_2 has two unsafe edges), then part (i) of the claim holds, since either $\delta(A \cap B), \delta(B \setminus A)$ each have two (or more) unsafe edges or else $\delta(A \cup B), \delta(A \setminus B)$ each have two (or more) unsafe edges. Similarly, if two of the unsafe edges of $\delta(B)$ are in the same set (i.e., if ϕ_3 or ϕ_4 has two unsafe edges), then part (i) of the claim holds. There is one remaining case: each of the four sets $F(A \cap B, B \setminus A), F(A \setminus B, V \setminus (A \cup B)), F(A \cap B, A \setminus B), F(B \setminus A, V \setminus (A \cup B))$ has exactly one unsafe edge. In this case, each of the four sets $\delta(A \cup B), \delta(A \cap B), \delta(A \setminus B), \delta(B \setminus A)$ has two unsafe edges. This proves part (i) of the claim. To prove part (ii) of the claim, observe that neither of the $(p+1)$ -cuts $\delta(A \cap B)$ or $\delta(A \setminus B)$ can have two (or more) unsafe edges, otherwise, a proper subset of the minimal violated set A would be violated. Then, the above case analysis shows that each of the two sets $F(A \cap B, B \setminus A), F(A \setminus B, V \setminus (A \cup B))$ has exactly one unsafe edge, and the set $F(B \setminus A, V \setminus (A \cup B))$ has two (or more) unsafe edges. This proves part (ii) of the claim.

Clearly, the function f is a pliable function, by Claim C.4, part (i). Next, we argue that the function f satisfies property (γ) .

Let C be a minimal violated set and let S_1, S_2 be violated sets such that C crosses both S_1, S_2 and $S_1 \subsetneq S_2$. W.l.o.g. assume that $S_2 \setminus (S_1 \cup C)$ is non-empty. Since C crosses S_2 , the edge-set $F(S_2 \setminus C, V \setminus (S_2 \cup C))$ has size $(p+1)/2$, by Claim C.3. Moreover, by Claim C.4, part (ii), this edge-set contains two unsafe edges. Observe that $S_1 \cup C$ crosses S_2 ; to see this, note that C crosses S_2 so the sets $C \cap S_2, C \setminus S_2, V \setminus (C \cup S_2)$ are non-empty, and, by assumption, $S_2 \setminus (S_1 \cup C)$ is non-empty. Applying Claim C.3 to this pair of crossing sets, we see that the edge-set $F(S_2 \setminus (S_1 \cup C), V \setminus (S_2 \cup C))$ has size $(p+1)/2$. Then we have $F(S_2 \setminus C, V \setminus (S_2 \cup C)) = F(S_2 \setminus (S_1 \cup C), V \setminus (S_2 \cup C))$, because both edge-sets have the same size and one edge-set is a subset of the other edge-set. Hence, $F(S_2 \setminus (S_1 \cup C), V \setminus (S_2 \cup C))$ contains two unsafe edges. Finally, by Claim C.3, the cut $\delta(S_2 \setminus (S_1 \cup C))$ has size $(p+1)$. Since this cut has two (or more) unsafe edges, $S_2 \setminus (S_1 \cup C)$ is a violated set. This proves that the function f satisfies property (γ) .

This completes the proof of Lemma 6.1. \square

D Optimal Dual Solutions with Non-Laminar Supports

In this section, we describe an instance of the AugSmallCuts problem where none of the optimal dual solutions (to the dual LP given in (2.2), Section 2) have a laminar support. Recall that the connectivity requirement function f for the AugSmallCuts problem is pliable and satisfies property (γ) , as seen in the proof of Theorem 1.2.

Consider the graph $G = (V, E)$ (shown in Figure 4 below using solid edges) which is a cycle on 4 nodes 1, 2, 3, 4, in that order. Edge-capacities are given by $u_{12} = 3, u_{23} = 4, u_{34} = 2, u_{41} = 1$. The link-set (shown using dashed edges) is $L = \{12, 23, 34, 41\}$, which is a disjoint copy of E . Link-costs are given by $c_{12} = c_{23} = c_{34} = 1$ and $c_{41} = 2$.

Consider the AugSmallCuts instance that arises when we choose $\tilde{\lambda} = 6$. The family of small cuts (with capacity strictly less than $\tilde{\lambda}$) is given by $\bigcup_{S \in \mathcal{A}} \{S, V \setminus S\}$, where

$$\mathcal{A} = \{\{1\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3\}\}.$$

The associated pliable function f satisfies $f(S) = 1$ if and only if $S \in \mathcal{A}$ or $V \setminus S \in \mathcal{A}$ holds. Observe that f is *not* uncrossable since $f(\{1, 2\}) = 1 = f(\{2, 3\})$, but $f(\{1, 2\} \cap \{2, 3\}) = f(\{2\}) = 0$ and $f(\{2, 3\} \setminus \{1, 2\}) = f(\{3\}) = 0$. Also note that the minimal violated set $\{2, 3\}$ (w.r.t. $F = \emptyset$) crosses the violated set $\{1, 2\}$.

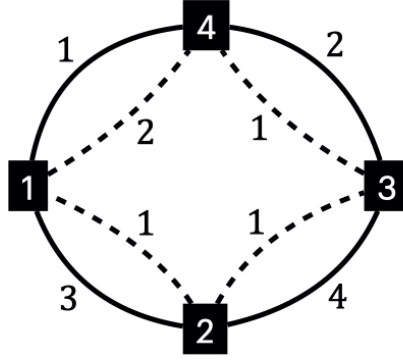


Fig. 4 An instance of the AugSmallCuts problem where every optimal dual solution has non-laminar support.

It can be seen that there are three inclusion-wise minimal link-sets that are feasible for the above instance and these are given by

$$\mathcal{C} := \{\{12, 23, 34\}, \{12, 41\}, \{34, 41\}\}.$$

Since each $F \in \mathcal{C}$ has cost 3, the optimal value for the instance is 3. Next, since L contains at least two links from every nontrivial cut, the vector $x \in [0, 1]^L$ with $x_e = \frac{1}{2}$, $\forall e \in L$ is a feasible augmentation for the fractional version of the instance, i.e., x is feasible for the primal LP given in (2.2), Section 2. Therefore, the optimal value of the primal LP is at most $\frac{5}{2}$.

Now, consider the dual LP, which is explicitly stated below. The dual packing-constraints are listed according to the following ordering of the links: 12, 23, 34, 41. For notational convenience, we use the shorthand y_1 to denote the dual variable $y_{\{1\}}$ corresponding to the set $\{1\}$. We use similar shorthand to refer to the dual variables of the other sets; thus, y_{234} refers to the dual variable $y_{\{2,3,4\}}$, etc.

$$\begin{array}{llllll} \max & (y_1 + y_{234}) & + & (y_{12} + y_{34}) & + & (y_{23} + y_{14}) & + & (y_{123} + y_4) \\ \text{subject to:} & (y_1 + y_{234}) & & & + & (y_{23} + y_{14}) & & \leq 1 \\ & & & (y_{12} + y_{34}) & & & & \leq 1 \\ & & & & & (y_{23} + y_{14}) & + & (y_{123} + y_4) \leq 1 \\ & (y_1 + y_{234}) & + & (y_{12} + y_{34}) & & & + & (y_{123} + y_4) \leq 2 \\ & & & & & & & y \geq 0. \end{array}$$

Observe that adding all packing constraints gives $2 \cdot \sum_{S \in \mathcal{A}} (y_S + y_{V \setminus S}) \leq 5$, hence, the optimal value of the dual LP is at most $5/2$. Moreover, a feasible dual solution with objective $5/2$ must satisfy the following conditions:

$$y_1 + y_{234} = y_{23} + y_{14} = y_{123} + y_4 = \frac{1}{2} \quad \text{and} \quad y_{12} + y_{34} = 1.$$

Clearly, there is at least one solution to the above set of equations, hence, by LP duality, the optimal value of both the primal LP and the dual LP is $5/2$.

Furthermore, any optimal dual solution y^* satisfies $\max(y_S^*, y_{V \setminus S}^*) > 0$ for all $S \in \mathcal{A}$ (by the above set of equations). We conclude by arguing that for any optimal dual solution y^* , its support $\mathcal{S}(y^*) = \{S \subseteq V : y_S^* > 0\}$ is non-laminar, because some two sets $A, B \in \mathcal{S}(y^*)$ cross. Since the relation A crosses B is closed under taking set-complements (w.r.t. the ground-set V), we may assume w.l.o.g. that the support contains each set in $\mathcal{A} = \{\{1\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3\}\}$. The support of y^* is not laminar because $\{1, 2\}$ and $\{2, 3\}$ cross.