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

Ishan Bansal
Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

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

Logan Grout
Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA

Sharat Ibrahimpur
Department of Mathematics, London School of Economics and Political Science, 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 a 16-approximation ratio via the primal-dual method 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 have a laminar support.

We present applications of our main result to three network-design problems.

1. A 16-approximation algorithm for augmenting the 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 $c \in \mathbb{Q}^E_{\geq 0}$ and edge capacities $u \in \mathbb{Z}^E_{\geq 0}$, find a minimum cost subset of the edges $F \subseteq E$ such that the capacity across any cut in (V, F) is at least k; u_{min} (respectively, u_{max}) denote 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.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

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

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.15

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2209.11209v2` [3]

Funding *Ishan Bansal*: Partially supported by Office of Naval Research (ONR) Grant N00014-21-1-2575.
Joseph Cheriyan: Supported in part by NSERC, RGPIN-2019-04197.

Sharat Ibrahimpur: 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.

Acknowledgements 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.

1 Introduction

The primal-dual method is a well-known algorithmic discovery of the past century. Kuhn (1955) [25] presented a primal-dual algorithm for weighted bipartite matching, and Dantzig et al. (1957) [9] 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 does not require solving the underlying LPs, which makes it attractive for both theoretical studies and real-world applications. Several 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 [19, Section 4.9], [16], [21], [27], [32].

Several decades after the pioneering work of Kuhn, Dantzig et al., the design of approximation algorithms for NP-hard problems emerged as an important area of research. Agrawal, Klein and Ravi [2] designed and analyzed a primal-dual approximation algorithm for the Steiner forest problem. Goemans and Williamson [18] generalized these algorithms to constrained forest problems. Subsequently, Williamson, Goemans, Vazirani and Mihail [33] (abbreviated WGMV) extended the methods of [18] 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 network-design problems.

Network design encompasses a wide class of problems that find applications in sectors like transportation, facility location, information security, and resource connectivity, to name a few. Due to its wide scope and usefulness, the area has been studied for decades and it has led to major algorithmic innovations. Most network-design problems are NP-Hard, and oftentimes even APX-hard, so researchers in the area have focused on designing 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 [26], Vazirani [31], and Williamson & Shmoys [34]. This property has been leveraged in various ways to obtain $O(1)$ approximation ratios for problems such as survivable network design [20], min-cost/min-size k-edge connected spanning subgraph [15, 14], min-cost 2-node connected spanning subgraph [11], $(p, 1)$-flexible graph connectivity [5], 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, most known techniques for designing $O(1)$ approximation algorithms fail to work. Indeed, only logarithmic approximation ratios are known for some of the problems where the uncrossability property does not hold. These logarithmic approximation ratios are usually obtained via a reduction to the set cover problem, for which a greedy strategy yields a logarithmic approximation. WGMV [33] 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 in this work is a novel analysis of the WGMV primal-dual approximation algorithm applied to a class of functions that strictly contain the class of uncrossable functions; we show that the algorithm still yields an $O(1)$ approximation guarantee for this larger class. This new class of functions captures some well-studied network design problems. An application of our main result provides improved approximation ratios for the capacitated k-edge connected subgraph problem, some instances of the flexible graph connectivity problem, and the problem of augmenting all *small* cuts of a graph. A detailed discussion of our results can be found in Section 1.1. For the benefit of the reader, in Section 2.1 we give an overview of WGMV's primal-dual algorithm and its analysis.

The primal-dual algorithm for solving network design problems follows the common strategy of starting with a graph that has no edges and then iteratively buying (i.e., including) a subset of edges into the infeasible solution until feasibility is attained. Within each iteration, the algorithm's goal is to buy a cheap edge-set that fixes some or all of the infeasibility of the current solution. Let F denote the edge-set that has been bought until some step in the algorithm. 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 prespecified connectivity requirement of S. The algorithm deems an edge to be useful if it is in the cut of a violated set S. Clearly, the family of violated sets is important for the design and analysis of these algorithms, especially the inclusion-wise minimal violated sets. A family $\mathcal{F}$ of sets is called *uncrossable* if the following holds:

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

Informally speaking, the uncrossability property ensures that the the minimal sets within the family can be considered independently. Formally, a minimal violated set A in an uncrossable family $\mathcal{F}$ cannot cross another set $S \in \mathcal{F}$; otherwise, we get a contradiction since $A, S \in \mathcal{F}$ implies that either $A \cap S$ or $A \setminus S$ is in $\mathcal{F}$. This key property is one of the levers used in the design of $O(1)$-approximation algorithms for some network-design problems. Unfortunately, there are important problems in network design where the family of violated sets does not form an uncrossable family. For instance, see the instance described in Appendix B. This leads us to define a new class of set families that contains all uncrossable families.

Call a family $\mathcal{F}$ *pliable* if the following holds:

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

In the full version of our paper, we show that the WGMV primal-dual algorithm has a super-constant approximation ratio for pliable families. Nevertheless, by enforcing an additional property on the given pliable family, we can establish that the WGMV algorithm yields an $O(1)$ approximation. We call this additional assumption property (γ); see Section 1.1.1 for the formal definition. From a structural standpoint, this property still allows a minimal violated set to cross another violated set, but, crucially, it does not allow them to cross an arbitrary number of violated sets in arbitrary ways. As we show later, the fact that disparate network design problems can be captured by pliable families with property (γ) hints that this property is "just right".

The above connectivity augmentation problems can be understood in a general framework called f-connectivity. In this problem, we are given an undirected graph $G = (V, E)$ on n vertices with nonnegative costs $c \in \mathbb{Q}^E_{\geq 0}$ on the edges and a requirement function $f : 2^V \to \{0, 1\}$ on subsets of vertices. 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 binary variables x_e model inclusion of edge e in F:

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

We remark that in its most-general form, f-connectivity is hard to approximate within a logarithmic factor. This can be shown via a reduction from the hitting set problem.[1] Thus, research on f-connectivity has focused on instances where f has some nice structural properties.

▶ **Definition 1** ([33]). *A function $f : 2^V \to \{0, 1\}$ 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 2.** *A function $f : 2^V \to \{0, 1\}$ satisfying $f(V) = 0$ is called* pliable *if for any $A, B \subseteq V$ with $f(A) = f(B) = 1$, we have $f(A \cap B) + f(A \cup B) + f(A \setminus B) + f(B \setminus A) \geq 2$.*

Note that the problem of augmenting an uncrossable (pliable) family can be seen as an f-connectivity problem whose requirement function is an uncrossable (pliable) function.

1.1 Our Contributions

In this work, we introduce the class of pliable functions and study the approximation ratio of WGMV's algorithm on f-connectivity instances arising from pliable functions. To the best of our knowledge, we are the first to investigate the f-connectivity problem beyond uncrossable functions. As mentioned before, the algorithm of WGMV can perform poorly on an arbitrary instance with a pliable function f. In the full version [3, Section 6], we present an instance where the solution returned by the WGMV algorithm costs $\Omega(\sqrt{n})$ times the optimal cost.

1.1.1 Pliable Functions and Property (γ)

As alluded to in the introduction, 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 ([33, Lemma 5.1(3)]). This property does not hold when we apply the primal-dual method to augment a pliable function; see the instance described in Appendix B. Nevertheless, we carve out a subclass of pliable functions – still containing all uncrossable functions – for which the WGMV algorithm yields an $O(1)$-approximate solution. This subclass is characterized by the following structural property that allows for minimal violated sets to cross other violated sets, but in a limited way.

[1] Given a ground-set X and a family $\mathcal{S}$ of subsets of X to hit, we define $L := \{\ell_x : x \in X\}$, $R := \{r_x : x \in X\}$, and $E := \{e_x = \ell_x r_x : x \in X\}$. We then take $G = (L \sqcup R, E)$ to be a bipartite graph with a perfect matching E and f to be the indicator function of the family $\{\{\ell_x : x \in A\} : A \in \mathcal{S}\}$.

Property (γ) : For any edge-set $F \subseteq E$ and for any violated sets (w.r.t. f and F) C, S_1, S_2, with $S_1 \subsetneq S_2$, the following conditional proposition holds:

$$(C \text{ is inclusion-wise minimal) and } (C \text{ crosses both } S_1 \text{ and } S_2)$$
$$\implies S_2 \setminus (S_1 \cup C) \text{ is either empty or violated.}$$

▶ **Theorem 3.** *Let $G = (V, E)$ be an undirected graph with nonnegative costs $c : E \to \mathbb{Q}_{\geq 0}$ on its edges, and let $f : 2^V \to \{0, 1\}$ 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 introduce the network-design applications where Theorem 3 gives new/improved approximation algorithms. In each of these applications, we setup an f-connectivity problem where the function f is a pliable function with property (γ).

1.1.2 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. Formally, in an instance of the AugSmallCuts problem we are given an undirected capacitated graph $G = (V, E)$ with edge-capacities $u \in \mathbb{Q}^E_{\geq 0}$, a set of links $L \subseteq \binom{V}{2}$ with costs $c \in \mathbb{Q}^L_{\geq 0}$, and a threshold $\widetilde{\lambda} \in \mathbb{Q}_{\geq 0}$. 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 $u(\delta(S) \cap E) < \widetilde{\lambda}$.

We remark that 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, $\lambda(G) := \min\{u(\delta(S) \cap E) \,:\, \emptyset \subsetneq S \subsetneq V\}$. Assuming u is integral and $\widetilde{\lambda} = \lambda(G) + 1$, we get the well-known connectivity augmentation problem for which constant-factor approximation algorithms are known [13, 23]. On the other hand, when $\widetilde{\lambda} = \infty$, 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 $\widetilde{\lambda}$. The proof of the following theorem is given in Section 4.

▶ **Theorem 4.** *There is a* 16*-approximation algorithm for the* AugSmallCuts *problem.*

As an aside, we refer the reader to Benczur & Goemans [4] 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.

In Appendix B, we give a small instance of the AugSmallCuts problem that illustrates some of the technical challenges which arise while working with the f-connectivity problem for a pliable function with property (γ). The instance described has bizarre properties that do not arise when working with uncrossable functions. First, it has a minimal violated set which crosses another violated set. Second, none of the optimal solutions to the dual LP of the f-connectivity problem are supported on a laminar family. The latter was believed to be a major hindrance to developing constant-factor approximation algorithms for general network-design problems.

1.1.3 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 $c \in \mathbb{Q}^E_{\geq 0}$ and edge capacities $u \in \mathbb{Z}^E_{\geq 0}$. 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 $u_{max} \leq k$.

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 [24] presented a combinatorial 2-approximation algorithm and Jain [20] matched this approximation guarantee via the iterative rounding method.

Goemans et al. [17] gave a $2k$-approximation algorithm for the general Cap-k-ECSS problem. Chakrabarty et al. [6] 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. [5] improved on these results by providing a $\min(k,\, 2u_{max})$-approximation algorithm. In this work, we give a $(16 \cdot \lceil k/u_{\min} \rceil)$-approximation algorithm, which leads to improved approximation guarantees when both $u_{\min}$ and $u_{\max}$ are sufficiently large. In particular, in the regime when $k \geq u_{\max} \geq u_{\min} \geq 32$ and $u_{\min} \cdot u_{\max} \geq 16k$.

▶ **Theorem 5.** *There is a $16 \cdot \lceil k/u_{min} \rceil$-approximation algorithm for the Cap-k-ECSS problem.*

The proof of Theorem 5 can be found in Section 5.

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

Adjiashvili, Hommelsheim and Mühlenthaler [1] introduced the model of Flexible Graph Connectivity that we denote by FGC. Boyd, Cheriyan, Haddadan and Ibrahimpur [5] 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 $c \in \mathbb{Q}^E_{\geq 0}$. A subset $F \subseteq E$ of edges is feasible for the (p, q)-FGC problem if for any set F' consisting of at most q unsafe edges, the subgraph $(V, F \setminus F')$ 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. [5] presented a 4-approximation algorithm for $(p, 1)$-FGC based on the WGMV primal-dual method, and they gave an $O(q \log n)$-approximation algorithm for general (p, q)-FGC and a $(q + 1)$-approximation for $(1, q)$-FGC. Concurrently with our work, Chekuri and Jain [8] obtained $O(p)$-approximation algorithms for $(p, 2)$-FGC, $(p, 3)$-FGC and $(2p, 4)$-FGC; in particular, they present a $(2p + 4)$-approximation ratio for $(p, 2)$-FGC. Chekuri and Jain have several other results for network design in non-uniform fault models; [7] have results on the flexible graph connectivity problem that arises from the classical survivable network design problem, which they call (p, q)-Flex-SNDP.

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

▶ **Theorem 6.** *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 [8], Theorem 6 yields a better approximation ratio when $p > 8$ or $p \in \{2, 4, 6, 8\}$. For $p = 1$, the approximation ratio of 3 from [5] is better than the guarantees given by [8] and Theorem 6. The proof of Theorem 6 can be found in Section 6.

1.2 Related work

Goemans & Williamson [18] introduced the notion of proper functions with the motivation of designing approximation algorithms for problems in network design. They formulated several of these problems as the f-connectivity problem where f is a proper function. A symmetric function $f : 2^V \to \mathbb{Z}_{\geq 0}$ 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 [20] 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. We mention that there are examples of uncrossable functions that are not weakly supermodular, see [5].

2 Preliminaries

This section has definitions and preliminary results. Our notation and terms are consistent with [10, 30], and readers are referred to those texts for further information.

For a positive integer k, we use $[k]$ to denote the set $\{1, \ldots, k\}$. For a ground-set V and a subset S of V, the complement of S (w.r.t. V) is denoted $V \setminus S$. 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 the informal phrasing "we apply the primal-dual method to augment a pliable function" instead of the phrasing "we apply the primal-dual method to an f-connectivity problem where the function f is a pliable function".

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 $c \in \mathbb{R}^E_{\geq 0}$ 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 $\textsc{opt}(G)$ 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 $\textsc{opt}$ rather than $\textsc{opt}(G)$.

Let $G = (V, E)$ be any multi-graph, let $A, B \subseteq V$ be two disjoint node-sets, and let $F \subseteq V$ 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 $E(A, V \setminus A)$. 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.

We use the following observations.

▶ **Fact 7.** *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 $e \in \delta_F(B)$. ◀

For any function $f : 2^V \to \{0, 1\}$ 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 8.** *Let $f : 2^V \to \{0, 1\}$ be a pliable function and $F \subseteq E$ be an edge-set. Define the function $f' : 2^V \to \{0, 1\}$ 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 7. 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. ◀

2.1 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. [33] that achieves approximation ratio 2 for an f-connectivity problem where the function f is an uncrossable function.

▶ **Theorem 9** (Lemma 2.1 in [33]). *Let $f : 2^V \to \{0, 1\}$ 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

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

Dual LP

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

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 incident to active sets. 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 9 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:

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

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 any iteration is ≤ 2, and this forms the crux of the WGMV analysis.

We refer the reader to [19] for a detailed discussion of the primal-dual method for network design problems.

3 Extending the WGMV Primal-Dual Method to Pliable functions

In this section, we prove our main result, Theorem 3: we show that the primal-dual algorithm outlined in Section 2.1 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. [33] which was outlined in Section 2.1. 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, after the reverse delete stage.

For the remainder of this proof we assume that the iteration, and thus $\mathcal{C}$, is fixed. We define $H := \cup_{C\in\mathcal{C}}\delta_{F'}(C)$. (Informally speaking, H is the subset of F' that is relevant for the analysis of our fixed iteration.) Additionally, to ease notation when discussing a laminar family of sets, 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 which can be proved by the same arguments as in the proof of [33, Lemma 5.1].

▶ **Lemma 10.** *For any edge $e \in H := \cup_{C\in\mathcal{C}}\delta_{F'}(C)$, there exists a witness set $S_e \subseteq V$ with:*
(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 A.

▶ **Lemma 11.** *There exists a laminar family of witness sets.*

▶ **Lemma 12.** *The active sets in $\mathcal{C}$ are pair-wise disjoint.*

Proof. Suppose that two sets $C_1, C_2 \in \mathcal{C}$ intersect. Then due to property (i) 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. ◀

Let $\mathcal{L}$ be the laminar family of witness sets 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 $\mathcal{T}$. The edges of $\mathcal{T}$ are oriented away from the root; thus, $\mathcal{T}$ has an oriented edge (v_Q, v_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, $n_R + n_G + n_B = |\mathcal{T}| = |F'| + 1$. Let n_L denote the number of leaf nodes of $\mathcal{T}$.

▶ **Lemma 13.** *The following are true:*

(i) *Each leaf node of $\mathcal{T}$ is red.*

(ii) *We have $n_G \leq n_L \leq n_R$.*

Proof. The first claim follows by repeating the argument in [33, Lemma 5.3]. Next, by (i), we have $n_L \leq n_R$. Moreover, 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. ◀

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 3. Clearly, the theorem would follow from the inequality $\sum_{C \in \mathcal{C}} |\delta_{F'}(C)| \leq O(1) \cdot |\mathcal{C}|$; thus, we need to prove an upper bound of $O(|\mathcal{C}|)$ 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 14 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 15 (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 13). 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 14.** *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. ◀

▶ **Lemma 15.** *We have that:*
(i) *Any oriented path of $\mathcal{T} \setminus \{v_V\}$ with four nodes has at least one non-black node.*
(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 an oriented path of $\mathcal{T}$ that has four black nodes and that is not incident to the root v_V; let $v_{S_4} \to v_{S_3} \to v_{S_2} \to v_{S_1}$ be such an oriented path. Thus, $S_1 \subsetneq S_2 \subsetneq S_3 \subsetneq S_4$ are witness sets of $\mathcal{L}$. 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 $\delta(S_{i+1})$).

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

▷ Claim 16. 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 17. 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 16, 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 18. 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 18); 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 $\widetilde{C}$. Clearly, the mapping ψ from $\mathcal{C}$ to $\mathcal{L}$ maps the active set $\widetilde{C}$ to a set $S_{\widetilde{C}}$. Either $S_{\widetilde{C}} = S_4$ or else $S_{\widetilde{C}}$ is a subset of 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 13 and the sketch given below Lemma 13. In more detail, at the start of the second phase, each node of $\mathcal{T}$ has ≤ 2 tokens, by Lemma 14. 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. ◀

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 4. 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 $u \in \mathbb{Q}^E_{\geq 0}$, a set of links $L \subseteq \binom{V}{2}$ with costs $c \in \mathbb{Q}^L_{\geq 0}$, and a threshold $\widetilde{\lambda} \in \mathbb{Q}_{\geq 0}$. 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 $u(\delta_E(S)) < \widetilde{\lambda}$.

Proof of Theorem 4. Define $f : 2^V \to \{0, 1\}$ such that $f(S) = 1$ if and only if $S \notin \{\emptyset, V\}$ and $u(\delta_E(S)) < \widetilde{\lambda}$. We apply Theorem 3 for the f-connectivity problem on the graph $H = (V, L)$ with edge-costs $c \in \mathbb{Q}^L_{\geq 0}$ 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 3 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 $2\widetilde{\lambda}$, 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 $f' : 2^V \to \{0, 1\}$ such that $f'(S) = 1$ if and only if $f(S) = 1$ and $\delta_F(S) = \emptyset$. By Fact 8, 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) $u(\delta_E(S_2 \setminus (S_1 \cup C))) < \widetilde{\lambda}$. Observe that S_2 crosses $(S_1 \cup C)$. To show (i), we apply Fact 7 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

$$\delta_E(S_2 \setminus (S_1 \cup C)) \,\cup\, \delta_E(C \setminus S_2) \text{ is a (multi-)subset of } \delta_E(S_2) \,\cup\, \delta_E(C \cup S_1).$$

(Note that for disjoint sets $A_1, A_2, A_3 \subsetneq V$, $\delta(A_1) \cup \delta(A_2)$ is a (multi-)subset of $\delta(A_1 \cup A_3) \cup \delta(A_2 \cup A_3)$.) Moreover, we claim that $u(\delta_E(C \cup S_1)) < \widetilde{\lambda}$ and $u(\delta_E(C \setminus S_2)) \geq \widetilde{\lambda}$. The two claims immediately imply (ii) (since $u(\delta_E(S_2)) < \widetilde{\lambda}$).

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 7, $\delta_F(C \cup S_1) = \emptyset$, hence, $f(C \cup S_1) = 1$; equivalently, $u(\delta_E(C \cup S_1)) < \widetilde{\lambda}$. Since C is a minimal violated set, $f'(C \setminus S_2) = 0$. Moreover, $\delta_F(C \setminus S_2) = \emptyset$, by Fact 7. Hence, $f(C \setminus S_2) = 0$; equivalently, $u(\delta_E(C \setminus S_2)) \geq \widetilde{\lambda}$.

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 $\widetilde{\lambda}$ 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 $\{\emptyset \subsetneq S \subsetneq V : u(\delta_{E'}(S)) < \widetilde{\lambda},\ u(\delta_{E'}(A)) \geq \widetilde{\lambda}\ \forall\ \emptyset \subsetneq A \subsetneq S\}$. We use the notion of solid sets to find all such minimally violated sets; see Naor, Gusfield, and Martel [29] and see Frank's book [12]. A solid set of an undirected graph $H = (V, E'')$ with capacities $w \in \mathbb{R}^{E''}_{\geq 0}$ 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 [29] and [12, 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. ◀

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 5. Our algorithm is based on repeated applications of Theorem 4.

Recall the capacitated k-edge-connected subgraph problem (Cap-k-ECSS): we are given an undirected graph $G = (V, E)$ with edge costs $c \in \mathbb{Q}^E_{\geq 0}$ and edge capacities $u \in \mathbb{Z}^E_{\geq 0}$. 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 5. 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 4 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 4, $c(F') \leq 16 \cdot c(F^*)$. 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. ◀

We remark that our new result (Theorem 4) 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 [5]. 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. Let us call such cuts *deficient*. Augmenting all deficient 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. In other words, the indicator function $f : 2^V \to \{0, 1\}$ of the sets S such that $\delta(S)$ is a deficient cut could violate the definition of an uncrossable function.

▶ **Example 19.** 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 cuts $\delta(\{v_1, v_2\})$ and $\delta(\{v_2, v_3\})$ are deficient. Note that the function $f : \{v_1, v_2, v_3, v_4\} \to \{0, 1\}$ that has $f(\{v_1, v_2\}) = f(\{v_2, v_3\}) = f(\{v_1\}) = f(\{v_4\}) = 1$ and $f(S) = 0$ for all other $S \subseteq V$ is not uncrossable (observe that the cuts $\delta(v_2)$ and $\delta(v_3)$ are not deficient). Moreover, observe that the minimal violated set $C = \{v_2, v_3\}$ crosses the violated set $S = \{v_1, v_2\}$.

Proof of Theorem 6. 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 $|\delta_F(S) \setminus \{e, f\}| \leq p - 1$. Since F is feasible for $(p, 1)$-FGC, we have $|\delta(S) \setminus \{e\}| \geq p$ and $|\delta(S) \setminus \{f\}| \geq p$. Thus, $|\delta_F(S)| = p + 1$. 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. Augmenting all such violated sets by at least one (safe or unsafe) edge will result in a feasible solution to $(p, 2)$-FGC. Let $f : 2^V \to \{0, 1\}$ 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 3.

Our proof of the following key lemma is presented in [3, Section 5].

▶ **Lemma 20.** *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, $\{S \subsetneq V \ : \ p \leq |\delta(S)| \leq 2p\}$ 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 [22, 28]. 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 [33] based on our results in Section 3. Furthermore, for even p, the approximation ratio is $6 (= 4 + 2)$ since the additive approximation-loss for the augmenting step is 2 when f is uncrossable (see Theorem 9). This completes the proof of Theorem 6. ◀

References

1. David Adjiashvili, Felix Hommelsheim, and Moritz Mühlenthaler. Flexible Graph Connectivity. *Mathematical Programming*, 192:409–441, 2022. doi:10.1007/s10107-021-01664-9.
2. Ajit Agrawal, Philip Klein, and R Ravi. When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks. *SIAM Journal on Computing*, 24(3):440–456, 1995. doi:10.1137/S0097539792236237.
3. Ishan Bansal, Joseph Cheriyan, Logan Grout, and Sharat Ibrahimpur. Improved Approximation Algorithms by Generalizing the Primal-Dual Method Beyond Uncrossable Functions. *CoRR*, abs/2209.11209v2, 2022. arXiv:2209.11209.
4. András A. Benczúr and Michel X. Goemans. Deformable Polygon Representation and Near-Mincuts. In *Building Bridges. Bolyai Society Mathematical Studies*, pages 103–135. Springer, 2008. doi:10.1007/978-3-540-85221-6_3.
5. Sylvia C. Boyd, Joseph Cheriyan, Arash Haddadan, and Sharat Ibrahimpur. Approximation algorithms for flexible graph connectivity. *Mathematical Programming*, 2023. doi:10.1007/s10107-023-01961-5.
6. Deeparnab Chakrabarty, Chandra Chekuri, Sanjeev Khanna, and Nitish Korula. Approximability of Capacitated Network Design. *Algorithmica*, 72(2):493–514, 2015. doi:10.1007/s00453-013-9862-4.
7. Chandra Chekuri and Rhea Jain. Approximating Flexible Graph Connectivity via Räcke Tree based Rounding. *CoRR*, abs/2211.08324, 2022. doi:10.48550/arXiv.2211.08324.
8. Chandra Chekuri and Rhea Jain. Augmentation based Approximation Algorithms for Flexible Network Design. *CoRR*, abs/2209.12273, 2022. doi:10.48550/arXiv.2209.12273.
9. George B. Dantzig, Lester R. Ford Jr., and Delbert R. Fulkerson. A Primal-Dual Algorithm for Linear Programs. In *Linear Inequalities and Related Systems*, volume 38 of *Annals of Mathematics Studies*, pages 171–182. Princeton University Press, 1957. doi:10.1515/9781400881987-008.
10. Reinhard Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer, 2017. doi:10.1007/978-3-662-53622-3.
11. Lisa Fleischer, Kamal Jain, and David P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences*, 72(5):838–867, 2006. doi:10.1016/j.jcss.2005.05.006.

12 András Frank. *Connections in Combinatorial Optimization*, volume 38 of *Oxford Lecture Series in Mathematics and Its Applications*. Oxford University Press, 2011.

13 Greg N. Frederickson and Joseph F. JáJá. Approximation Algorithms for Several Graph Augmentation Problems. *SIAM Journal on Computing*, 10(2):270–283, 1981. doi:10.1137/0210019.

14 Harold N. Gabow and Suzanne Gallagher. Iterated Rounding Algorithms for the Smallest k-Edge Connected Spanning Subgraph. *SIAM Journal on Computing*, 41(1):61–103, 2012. doi:10.1137/080732572.

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

16 Harold N. Gabow, Michel X. Goemans, and David P. Williamson. An efficient approximation algorithm for the survivable network design problem. *Mathematical Programming*, 82:13–40, 1998. doi:10.1007/BF01585864.

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

18 Michel X. Goemans and David P. Williamson. A General Approximation Technique for Constrained Forest Problems. *SIAM Journal on Computing*, 24(2):296–317, 1995. doi:10.1137/S0097539793242618.

19 Michel X. Goemans and David P. Williamson. The Primal-Dual Method for Approximation Algorithms and Its Application to Network Design Problems. In *Approximation Algorithms for NP-Hard Problems*, chapter 4, pages 144–191. PWS Publishing Company, 1996. URL: https://math.mit.edu/~goemans/PAPERS/book-ch4.pdf.

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

21 David S. Johnson, Maria Minkoff, and Steven Phillips. The Prize Collecting Steiner Tree Problem: Theory and Practice. In David B. Shmoys, editor, *Proceedings of the 11th Symposium on Discrete Algorithms*, pages 760–769, 2000. URL: https://dl.acm.org/doi/10.5555/338219.338637.

22 David R. Karger and Clifford Stein. A New Approach to the Minimum Cut Problem. *Journal of the ACM*, 43(4):601–640, 1996. doi:10.1145/234533.234534.

23 Samir Khuller and Ramakrishna Thurimella. Approximation Algorithms for Graph Augmentation. *Journal of Algorithms*, 14(2):214–225, 1993. doi:10.1006/jagm.1993.1010.

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

25 Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. doi:10.1002/nav.3800020109.

26 Lap Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2011. doi:10.1017/CBO9780511977152.

27 Milena Mihail, David Shallcross, Nate Dean, and Marco Mostrel. A Commercial Application of Survivable Network Design: ITP/INPLANS CCS Network Topology Analyzer. In *Proceedings of the 7th Symposium on Discrete Algorithms*, pages 279–287, 1996. URL: https://dl.acm.org/doi/10.5555/313852.314074.

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

29 Dalit Naor, Dan Gusfield, and Charles Martel. A Fast Algorithm for Optimally Increasing the Edge Connectivity. *SIAM Journal on Computing*, 26(4):1139–1165, 1997. doi:10.1137/S0097539792234226.

30 Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.
31 Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2003. doi:10.1007/978-3-662-04565-7.
32 David P. Williamson and Michel X. Goemans. Computational Experience with an Approximation Algorithm on Large-Scale Euclidean Matching Instances. *INFORMS Journal on Computing*, 8(1):29–40, 1996. doi:10.1287/ijoc.8.1.29.
33 David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani. A Primal-Dual Approximation Algorithm for Generalized Steiner Network Problems. *Combinatorica*, 15(3):435–454, 1995. doi:10.1007/BF01299747.
34 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

A Missing Proofs from Section 3

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

▶ **Lemma 21.** *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 perform an exhaustive case analysis to check that the lemma is true. 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. 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 [33] can be applied.
2. If $S_1 \cup S_2$ and $S_1 \setminus S_2$ are violated, then consider where the end-nodes of the edges e_1 and e_2 lie. If $e_1 \in E(S_1 \setminus S_2, V \setminus (S_1 \cup S_2))$ and $e_2 \in E(S_1 \setminus S_2, S_1 \cap S_2)$, then we can set $S_1' = S_1 \cup S_2$ and $S_2' = S_2$. The other possibilities for e_1 and e_2 are handled similarly.
3. If $S_1 \cap S_2$ and $S_1 \setminus S_2$ are violated, again consider where the end-nodes of the edges e_1 and e_2 lie. If $e_1 \in E(S_1 \setminus S_2, V \setminus (S_1 \cup S_2))$ and $e_2 \in E(S_1 \setminus S_2, S_1 \cap S_2)$, then we can set $S_1' = S_1 \cap S_2$ and $S_2' = S_2$. The other possibilities for e_1 and e_2 are handled similarly.

This completes the proof of the lemma. ◀

▶ **Lemma 22.** *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. ◀

▶ **Lemma 11.** *There exists a laminar family of witness sets.*

Proof. We show that any witness family can be uncrossed and made laminar. We prove this by induction on the size of the witness family ℓ.

Base Case: Suppose $\ell = 2$, then one application of Lemma 21 is sufficient.

Inductive Hypothesis: If $S_1, \ldots, S_\ell$ are witness sets for edges $e_1, \ldots, e_\ell$ respectively with $\ell \leq k$, then, by repeatedly applying Lemma 21, one can construct witness sets $S'_1, \ldots, S'_\ell$ for the edges $e_1, \ldots, e_\ell$ respectively such that $S'_1, \ldots, S'_\ell$ is a laminar family.

Inductive Step: Consider $k+1$ witness sets $S_1, \ldots, S_{k+1}$. By the inductive hypothesis, we can uncross all the witness sets $S_1, \ldots, S_k$ to obtain witness sets $S'_1, \ldots, S'_k$ 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 $S'_2, \ldots, S'_k, S_{k+1}$ and we obtain a laminar family of witness sets, none of which overlap S'_1 either (by Lemma 22) and so we are done.
2. Suppose S_{k+1} overlaps all the sets $S'_1, \ldots, S'_k$ and for some S'_i, say S'_1, applying Lemma 21 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 $S'_2, \ldots, S'_{k+1}$, hence, applying the inductive hypothesis to these k sets gives us a laminar family of witness sets $S''_2, \ldots, S''_k$. By Lemma 22, S'_1 does not overlap any of the sets $S''_2, \ldots, S''_k$ and so we are done.
3. Suppose S_{k+1} overlaps all the sets $S'_1, \ldots, S'_k$ and, for every S'_i, applying Lemma 21 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 $S''_1, \ldots, S''_k, S_{k+1}$ with the property that S_{k+1} does not overlap any of the other sets. Applying the inductive hypothesis to the k sets $S''_1, \ldots, S''_k$ gives us a laminar family of witness sets $S'''_1, \ldots, S'''_k$. By Lemma 22, S_{k+1} does not overlap any of the sets $S'''_1, \ldots, S'''_k$ and so we are done. ◀

B 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 (2.1), 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 4.

Consider the graph $G = (V, E)$ (shown in Figure 1 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 $L = \{12, 23, 34, 41\}$, 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 $\widetilde{\lambda} = 6$. The family of small cuts (with capacity strictly less than $\widetilde{\lambda}$) 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 $\{1, 2\}$.

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

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

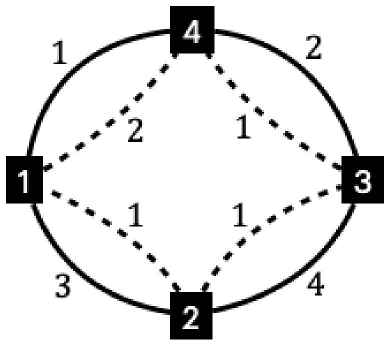

Figure 1 An instance of the AugSmallCuts problem where none of the optimal dual solutions have a laminar support.

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 (2.1), 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.

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

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.