Node-weighted Network Design in Planar and Minor-closed Families of Graphs

CHANDRA CHEKURI, University of Illinois
ALINA ENE, Boston University
ALI VAKILIAN, Toyota Technological Institute at Chicago (TTIC)

We consider *node-weighted* survivable network design (SNDP) in planar graphs and minor-closed families of graphs. The input consists of a node-weighted undirected graph $G = (V, E)$ and integer connectivity requirements $r(uv)$ for each unordered pair of nodes uv. The goal is to find a minimum weighted subgraph H of G such that H contains $r(uv)$ disjoint paths between u and v for each node pair uv. Three versions of the problem are edge-connectivity SNDP (EC-SNDP), element-connectivity SNDP (Elem-SNDP), and vertex-connectivity SNDP (VC-SNDP), depending on whether the paths are required to be edge, element, or vertex disjoint, respectively. Our main result is an $O(k)$-approximation algorithm for EC-SNDP and Elem-SNDP when the input graph is planar or more generally if it belongs to a proper minor-closed family of graphs; here, $k = \max_{uv} r(uv)$ is the maximum connectivity requirement. This improves upon the $O(k \log n)$-approximation known for node-weighted EC-SNDP and Elem-SNDP in general graphs [31]. We also obtain an $O(1)$ approximation for node-weighted VC-SNDP when the connectivity requirements are in $\{0, 1, 2\}$; for higher connectivity our result for Elem-SNDP can be used in a black-box fashion to obtain a logarithmic factor improvement over currently known general graph results. Our results are inspired by, and generalize, the work of Demaine, Hajiaghayi, and Klein [13], who obtained constant factor approximations for node-weighted Steiner tree and Steiner forest problems in planar graphs and proper minor-closed families of graphs via a primal-dual algorithm.

CCS Concepts: • **Theory of computation → Routing and network design problems**;

Additional Key Words and Phrases: Approximation algorithms, network design, primal-dual algorithms, element/vertex connectivity

1 INTRODUCTION

Network design is an important area of discrete optimization with several practical applications. Moreover, the clean optimization problems that underpin the applications have led to fundamental theoretical advances in combinatorial optimization, algorithms, and mathematical programming. In this article, we consider a class of problems that can be modeled as follows: Given an undirected graph $G = (V, E)$ find a subgraph H of *minimum weight/cost* such that H satisfies certain desired *connectivity* properties. A common cost model is to assign a non-negative weight $w(e)$ to each $e \in E$ and the weight of H is simply the total weight of edges in it. A number of well-studied problems can be cast as special cases. Examples include polynomial-time solvable problems such as the **minimum weight spanning tree (MST)** problem when H is required to connect all the nodes of G, and the **NP**-hard Steiner Tree problem where H is required to connect only a given subset $S \subseteq V$ of terminals. A substantial generalization of these problems is the ***survivable network design problem*** **(SNDP)**, which is defined as follows: The input, in addition to G, consists of an integer requirement function $r(uv)$ for each (unordered) pair of nodes uv in G; the goal is to find a minimum-weight subgraph H such that it contains, for each pair of nodes uv, $r(uv)$ disjoint-paths between u and v. We obtain two fundamental variants: If the $r(uv)$ paths for uv are required to be edge-disjoint, then it is called **edge-connectivity SNDP problem (EC-SNDP)**, and if they are required to be internally vertex-disjoint the problem is called **vertex-connectivity SNDP (VC-SNDP)**. These problems are relevant in designing fault-tolerant networks. It is not hard to see that VC-SNDP is a generalization of EC-SNDP. Moreover, VC-SNDP is known to be strictly harder than EC-SNDP from an approximation point of view. A problem of intermediate complexity is the **element-connectivity SNDP problem (Elem-SNDP)**: Here, the vertex set V is partitioned into *reliable nodes* R and *non-reliable nodes* $V \setminus R$. The requirements are only between terminal nodes $T \subseteq R$. The goal is to find a subgraph H of minimum weight such that each pair of terminals uv has $r(uv)$ element-disjoint paths, that is, paths that are disjoint in edges and non-reliable nodes. The problems mentioned so far arise naturally in concrete applications. Algorithmic approaches for these problems are in fact based on solving a larger class of abstract network design problems such as covering proper and skew-supermodular cut-requirement functions that we describe formally later.

Node weights: The cost of a network is dependent on the application. In connectivity problems, as we remarked, a common model is the edge-weight model. A more general problem is defined when each node v of G has a weight $w(v)$ and the weight of H is the total weight of the nodes in H.[1] Node weights are relevant in several applications, in particular telecommunication networks, where they can model the cost of setting up routing and switching infrastructures at a given node. There have also been several recent applications in wireless network design [33, 35] where the weight function is closely related to that of node weights. We refer the reader to [13] for some additional applications of node weights to network formation games.

The node-weighted versions of network design problems often turn out to be strictly harder to approximate than their corresponding edge-weighted versions. For instance, the Steiner Tree problem admits a 1.39-approximation for edge-weights [6]; however, Klein and Ravi [26] showed via a simple reduction from the Set Cover problem that the node-weighted Steiner Tree problem on n nodes is hard to approximate to within an $\Omega(\log n)$-factor unless **P = NP**. They also described a $(2 \log m)$-approximation where m is the number of terminals. A more dramatic difference emerges for SNDP. While Jain gave a 2-approximation for EC-SNDP with edge-weights

---

[1]For many problems of interest including Steiner Tree and SNDP the version with weights on both edges and nodes can be reduced to the version with only node weights; simply sub-divide an edge e by placing a new node v_e and place the weight of e on v_e.

Table 1. Approximation Ratios for SNDP and Related Problems

	Edge Weighted Graphs		Node Weighted Graphs	
	General	Planar	General	Planar
Steiner Tree	1.39 [7]	PTAS [5]	$O(\log n)$ [26]	2.4 [4]
Steiner Forest	2 [1]	PTAS [3]	$O(\log n)$ [26]	2.4 [4]
{0, 1}-Proper Functions	2 [19]	2 [19]	$O(\log n)$ [26]	6 [13]
EC-SNDP	2 [24]	2 [24]	$O(k \log n)$ [31]	$10k$
Elem-SNDP	2 [15]	2 [15]	$O(k \log n)$ [34]	$10k$
{0, 1, 2}-VC-SNDP	2 [15]	2 [15]	$O(\log n)$ [31]	13
VC-SNDP	$O(k^3 \log n)$ [11]	$O(k^3 \log n)$ [11]	$O(k^4 \log^2 n)$ [11]	$O(k^4 \log n)$

The entries with no citation are from this article. There is an $\Omega(\log n)$-hardness for all the node-weighted problems in the table for general graphs. The term k refers to the maximum connectivity requirement in the given instance.

[24], the best known approximation guarantee for EC-SNDP with node-weights is $O(k \log n)$ [31] where $k = \max_{uv} r(uv)$ is the maximum connectivity requirement. Moreover, Nutov [31] gives evidence, via a reduction from the k-Densest-Subgraph problem, that for the node-weighted problem a dependence on k in the approximation ratio is necessary. Table 1 summarizes the known approximation ratios and hardness results for some of the main problems considered in this article. One notices that the node-weighted version of problems have at least a logarithmic factor worse approximation than the corresponding edge-weighted problem.

Demaine, Hajiaghayi, and Klein [13] considered the approximability of the node-weighted Steiner Tree problem in *planar* graphs. They were partly motivated by the goal of overcoming the $\Omega(\log n)$-hardness that holds in general graphs. They described a primal-dual algorithm that is adapted from the well-known algorithm for the edge-weighted case [1, 19] and showed that it gives a 6-approximation in planar graphs. Demaine et al. also showed that their algorithm works for a more general class of {0, 1}-proper functions (first considered by Goemans and Williamson [19]) that includes several other problems such as the Steiner Forest problem. Their analysis shows that one obtains a constant factor approximation (the algorithm is the same) for any proper minor-closed family of graphs where the constant depends on the family. In addition to their theoretical values, these results have the potential to be useful in practice; the algorithm is simple and efficient to implement, and it is reasonable to assume that real-work networks that arise in several applications are close to being planar.

1.1 Our Results

In this article, we consider node-weighted network design problems in planar graphs for *higher connectivity*. In particular, we consider EC-SNDP, Elem-SNDP, and VC-SNDP and show that the basic insight in [13] can be built upon to develop improved approximation algorithms for these more general problems as well. Although we follow the high-level outline of [13], our results require susbtantial technical work. Our core result is for Elem-SNDP, which captures EC-SNDP as a special case and can be used in a black box fashion for VC-SNDP.

THEOREM 1.1. *There is a $10k$-approximation algorithm for node-weighted Elem-SNDP in planar graphs where k is the maximum requirement. Moreover, an $O(k)$ approximation guarantee also holds for graphs from a proper minor-closed family of graphs $\mathcal{G}$ where the constant in the approximation factor only depends on the family $\mathcal{G}$.*

Node-weighted EC-SNDP can be reduced easily in an approximation preserving fashion to node-weighted Elem-SNDP by choosing all nodes in the input graph to be reliable nodes. Thus, the preceding theorem applies to node-weighted EC-SNDP. Chuzhoy and Khanna [11] showed a generic

reduction of VC-SNDP to Elem-SNDP that does not change the underlying graph. Using Theorem 1.1 and the reduction in [11], we obtain the following corollary:

COROLLARY 1.2. *There is an $O(k^4 \log n)$-approximation for node-weighted VC-SNDP in proper minor-closed family of graphs.*

We also obtain an $O(1)$-approximation algorithm for node-weighted VC-SNDP when the requirements are in the set $\{0, 1, 2\}$.

THEOREM 1.3. *There is a 13–approximation algorithm for node-weighted VC-SNDP with $\{0, 1, 2\}$ connectivity requirements in planar graphs and, more generally, an $O(1)$-approximation for graphs from a proper minor-closed family of graphs.*

In summary, as mentioned in Table 1, our results show that provably better approximation guarantees can be obtained for node-weighted network design in planar graphs when compared to the case of general graphs.

1.2 Overview of Technical Ideas and Contribution

There are two main algorithmic approaches for SNDP. The first approach is the *augmentation* approach pioneered by Williamson et al. [37]. In this approach the desired network is built in k phases; at the end of the first $(\ell - 1)$ phases the connectivity of a pair uv is at least $\min\{r(uv), \ell - 1\}$. Thus, the optimization problem of ℓth phase is to increase the connectivity of certain pairs by *one*; the advantage is that we need to work with a 0-1 covering function. In the case of EC-SNDP, the augmentation problem is the problem of covering a skew-supermodular function. A requirement function $f : 2^V \to \mathbb{Z}_+$ is skew-supermodular[2] if for any $A, B \subseteq V$

$$f(A) + f(B) \leq \max\{f(A \cap B) + f(A \cup B), f(A \setminus B) + f(B \setminus A)\}.$$

Williamson et al. [37] showed that a primal-dual algorithm achieves a 2-approximation for covering edge-weighted 0-1 skew-supermodular functions. For the node-weighted variant, Nutov [31] gave an $O(\log n)$-approximation. These results for covering 0-1 skew-supermodular functions when combined with the augmentation give, respectively, $2k$ and $O(k \log n)$-approximation for the edge-weighted and node-weighted EC-SNDP.[3] For solving edge-weighted Elem-SNDP in the augmentation framework, [25] works with *skew-bisupermodular* functions that are more involved although the achieved approximation ratios are similar. The second approach for SNDP is the powerful *iterative rounding* technique pioneered by Jain that led to a 2-approximation for EC-SNDP [24] and Elem-SNDP [15]. Iterative rounding does not quite apply to node-weighted problems for a variety of technical reasons as well as known hardness of approximation results. For this reason, the main approach for attacking node-weighted SNDP problems has been the augmentation approach.

In this article, we follow the augmentation approach for node-weighted SNDP problems. Demaine et al. adapted the primal-dual algorithm for edge-weighted 0-1 proper functions to the node-weighted case. The novel technical ingredient in their analysis is to understand the properties of *node-minimal* feasible solutions instead of edge-minimal feasible solutions. The analysis crucially relies on the average degree of a planar graph being constant. For the most part, problems captured by 0-1 proper functions are very similar to the Steiner Forest problem, a canonical problem in this class. In this setting it is possible to visualize and understand node-minimal solutions

---

[2]This class of functions is also referred by other names such as uncrossable and weakly supermodular.
[3]The approximation for the edge-weighted version can be improved to $2H_k$ by doing the augmentation in the reverse order [18].

through connected components and basic reachability properties. In the augmentation approach for higher-connectivity, as we remarked, the problem in each phase is no longer a proper function but belongs to the richer class of skew-supermodular functions. The primal-dual analysis for this class of functions is more subtle and abstract and proceeds via uncrossing arguments and laminar witness families [37].

In a previous conference version of this work [8], we considered node-weighted EC-SNDP (and more general problems) in planar and minor-closed families of graphs. Based on properties of node-minimal feasible solutions for 0-1 skew-supermodular families, we obtained an $O(1)$-approximation. In this article, we extend the ideas to handle Elem-SNDP by considering node-minimal feasible solutions of 0-1-skew-bisupermodular functions (based on bisets) that arise in the augmentation framework for Elem-SNDP [10, 15, 25, 32]. An important and crucial aspect of the algorithm, which also applied to the results in [8], is that our results only apply for covering a restricted class of skew-bisupermodular functions that satisfy additional properties. We provide an example later to illustrate the reason why this is necessary.

As in [13], planarity plays a role only in one step of the analysis where we rely on the property that any minor[4] of the original graph has bounded average degree. Recall that planar graphs are a minor-closed family of graphs and the average degree is at most 6. This is the reason that the algorithm and analysis generalize to any proper minor-closed family of graphs. In the interest of clarity and exposition, we have not attempted to optimize the constants in the approximation analysis; the bound could perhaps be improved with a more careful analysis.

Other related work: Moldenhauer [30] showed that an improved analysis of the algorithm of Demaine et al. [13] reduces their 6-approximation guarantee of the node-weighted Steiner Forest in planar graphs to a bound of 3, which is tight for the algorithm. Moreover, he claimed, via a different algorithm, a 9/4-approximation. However, Berman and Yaroslavtsev [4] showed that the result of [30] suffers from a mistake in the analysis and that the correct approximation guarantee for the algorithm in [30] is 18/7. [4] developed an algorithm for the node-weighted planar Steiner Forest with a 2.4-approximation. [4, 30] point out the connection between node-weighted Steiner Forest and (Subset) Feedback Vertex Set in planar graphs for which Goemans and Williamson [21] had developed primal-dual algorithms—more details can be found in the papers.

Network design is a broad area that has been explored in depth over the years. We refer the reader to [20] for a survey on designing algorithms for network design problems using the primal-dual method, and to recent surveys cited in [23, 27] for an overview of the literature and references. We borrow several ideas from work on Elem-SNDP [10, 15, 25, 32] and VC-SNDP [11, 32]. Some variants of the SNDP problem such as Prize-collecting SNDP, Budgeted SNDP, and Network Activation have been studied in node-weighted setting and we refer readers to [2, 9, 17, 31, 36] for developments in these directions, several of which have happened after the conference version of this article appeared.

We also refer to some past and more recent work on other problems such as connected dominating set, connected vertex cover, and feedback vertex set that have been studied in the node-weighted setting [12, 14, 16, 22]. For several of these problems, the approximation ratios are better for graphs from proper minor-closed families compared to the ones for general graphs, and several admit a polynomial-time approximation scheme. The techniques in these papers are based on different lines of work and do not yet appear to apply to node-weighted Steiner tree problem, and our work here addresses higher connectivity problems.

---

[4]A minor of a graph G is another graph obtained by a sequence of edge and vertex deletions and edge contractions.

Organization: Section 2 sets up the relevant technical background on bisets and certain abstract properties that play a critical role in the analysis. Section 3 outlines the augmentation framework and the properties of the requirement function that arises in each phase of the augmentation framework. Section 4 (in particular, 4.3) describes the main technical result of this article, which is a constant factor approximation algorithm for the problem of covering node-weighted biuncrossable functions that arise in the augmentation framework. In Section 5, we apply the framework to handle {0, 1, 2}-VC-SNDP. We discuss some open problems and conclude in Section 6.

2 PRELIMINARIES

We formally define node-weighted EC-SNDP and Elem-SNDP. The input to node-weighted EC-SNDP is an undirected graph $G = (V, E)$, a weight function $w : V \to \mathbb{R}_+$, and a non-negative integer requirement $r(uv)$ for each unordered pair of nodes uv. The goal is to find a minimum weight subgraph $H = (V_H, E_H)$ such that H has $r(uv)$ edge-disjoint paths for each node pair uv; by the weight of H, we mean $w(V_H)$, since we are considering node-weights. Call a node u a terminal if it participates in a pair uv such that $r(uv) > 0$. Any feasible solution H contains all terminals and, hence, we can assume without loss of generality that the weight of terminals is zero.

The input to node-weighted Elem-SNDP is an undirected graph $G = (V, E)$ along with a partition of V into reliable nodes R and non-reliable nodes $V \setminus R$. The *elements* of G are $E \cup (V \setminus R)$. The input also specifies a non-negative weight function $w : V \to \mathbb{R}_+$ and integer requirements $r(uv)$ only over pairs uv where both u, v are *reliable* nodes. The goal is to find a minimum weight subgraph $H = (V_H, E_H)$ of G such that for each pair uv of reliable nodes, H has $r(uv)$ element-disjoint paths. For simplicity, we can assume that R forms an independent set in G by sub-dividing each edge uv where both $u, v \in R$ and adding a new non-reliable node. Then element-disjoint paths correspond to paths that are disjoint on non-reliable nodes. Once again, we can assume that any node u that participates in a pair uv such that $r(uv) > 0$ can be assumed to have zero weight, since it has to be included in every feasible solution. It is straightforward to see that EC-SNDP is a special case of Elem-SNDP in which all nodes in the input graph are reliable.

Following the general approach from prior work, we reduce SNDP to a more abstract problem of covering certain set and biset requirement functions. We set up the desired notation and definitions for this purpose and state several basic properties. We borrow extensively from past work [10, 15, 32] and give a few proofs here for the sake of completeness.

A key definition is that of *biset*. A biset is a pair of sets $\hat{S} = (S_{\text{in}}, S_{\text{out}}) \in 2^V \times 2^V$ such that $S_{\text{in}} \subseteq S_{\text{out}}$. The set S_{in} is the *inner part* of $\hat{S}$, S_{out} is the *outer part* of $\hat{S}$, and $S_{\text{out}} \setminus S_{\text{in}}$ is the *boundary* of $\hat{S}$ which is also denoted by $\text{bd}(\hat{S})$. We define the $\subseteq$ relation on the bisets as follows: We say $\hat{S} \subseteq \hat{T}$ iff $S_{\text{in}} \subseteq T_{\text{in}}$ and $S_{\text{out}} \subseteq T_{\text{out}}$. We use the terminology $\hat{S} \subset \hat{T}$ if $\hat{S} \subseteq \hat{T}$ and $\hat{S} \neq \hat{T}$. Furthermore, the union, intersection, and difference of $\hat{S}$ and $\hat{T}$ are defined as $\hat{S} \cap \hat{T} = (S_{\text{in}} \cap T_{\text{in}}, S_{\text{out}} \cap T_{\text{out}})$, $\hat{S} \cup \hat{T} = (S_{\text{in}} \cup T_{\text{in}}, S_{\text{out}} \cup T_{\text{out}})$, and $\hat{S} \setminus \hat{T} = (S_{\text{in}} \setminus T_{\text{out}}, S_{\text{out}} \setminus T_{\text{in}})$.

The following two propositions are straightforward to verify:

PROPOSITION 2.1. *The $\subseteq$ relation is a partial order over the bisets.*

PROPOSITION 2.2. $\hat{S} \cap \hat{T} \subseteq \hat{S}$ *and* $\hat{S} \setminus \hat{T} \subseteq \hat{S}$.

Definition 2.3 (Crossing Bifamily). A family of bisets $\mathcal{P}$ is *crossing* iff, for any bisets $\hat{S}$ and $\hat{T}$ in $\mathcal{P}$, union, intersection, and differences of $\hat{S}$ and $\hat{T}$ are in $\mathcal{P}$.

Definition 2.4 (Bimaximal Function). Let $\mathcal{P}$ be a crossing bifamily. A function $f : \mathcal{P} \to \mathbb{Z}$ is bimaximal iff, for any $\hat{S}, \hat{T} \in \mathcal{P}$ whose inner parts are disjoint (that is, $S_{\text{in}} \cap T_{\text{in}} = \emptyset$),

$$f(\hat{S} \cup \hat{T}) \leq \max\{f(\hat{S}), f(\hat{T})\}. \tag{1}$$

Definition 2.5 (Bisubmodular Function). Let $\mathcal{P}$ be a crossing bifamily. A function $f : \mathcal{P} \to \mathbb{Z}$ is bisubmodular iff for any $\hat{S}, \hat{T} \in \mathcal{P}$, both of the following inequalities hold[5]:

$$f(\hat{S}) + f(\hat{T}) \ge f(\hat{S} \cap \hat{T}) + f(\hat{S} \cup \hat{T}), \tag{2}$$

$$f(\hat{S}) + f(\hat{T}) \ge f(\hat{S} \setminus \hat{T}) + f(\hat{T} \setminus \hat{S}). \tag{3}$$

A function f is *bisupermodular* iff $-f$ is bisubmodular.

Definition 2.6 (Skew-bisupermodular Function). Let $\mathcal{P}$ be a crossing bifamily. A function $f : \mathcal{P} \to \mathbb{Z}$ is skew-bisupermodular iff for any $\hat{S}, \hat{T} \in \mathcal{P}$, one of the following holds:

$$f(\hat{S} \cap \hat{T}) + f(\hat{S} \cup \hat{T}) \ge f(\hat{S}) + f(\hat{T}), \tag{4}$$

$$f(\hat{S} \setminus \hat{T}) + f(\hat{T} \setminus \hat{S}) \ge f(\hat{S}) + f(\hat{T}). \tag{5}$$

Definition 2.7 (Biuncrossable Function). Let $\mathcal{P}$ be a crossing bifamily. A function $f : \mathcal{P} \to \mathbb{Z}$ is biuncrossable iff for any $\hat{S}, \hat{T} \in \mathcal{P}$ such that $f(\hat{S}) > 0$ and $f(\hat{T}) > 0$, one of the following holds:

$$f(\hat{S} \cap \hat{T}) + f(\hat{S} \cup \hat{T}) \ge f(\hat{S}) + f(\hat{T}), \tag{6}$$

$$f(\hat{S} \setminus \hat{T}) + f(\hat{T} \setminus \hat{S}) \ge f(\hat{S}) + f(\hat{T}). \tag{7}$$

PROPOSITION 2.8 (LEMMA 3.8 IN [15]). *Let f be a skew-bisupermodular function and let g be a bisubmodular function on the same domain. Then $f - g$ is a skew-bisupermodular function.*

PROPOSITION 2.9. *The $|\mathrm{bd}(.)|$ function is a bisubmodular function over the set of all bisets over V.*

The proof of Proposition 2.9 follows from the following propositions:

PROPOSITION 2.10. *For any two bisets $\hat{S}$ and $\hat{T}$, $|\mathrm{bd}(\hat{S})| + |\mathrm{bd}(\hat{T})| \ge |\mathrm{bd}(\hat{S} \cap \hat{T})| + |\mathrm{bd}(\hat{S} \cup \hat{T})|$.*

PROOF. Consider a vertex v that contributes to RHS. Then one of the following cases holds:

- $v \in \mathrm{bd}(\hat{S} \cap \hat{T}) \setminus \mathrm{bd}(\hat{S} \cup \hat{T})$. This implies that $v \in S_{\mathrm{out}} \cap T_{\mathrm{out}}$ and at least one of S_{in} and T_{in} does not contain v. Thus, $v \in \mathrm{bd}(\hat{S}) \cup \mathrm{bd}(\hat{T})$.
- $v \in \mathrm{bd}(\hat{S} \cup \hat{T}) \setminus \mathrm{bd}(\hat{S} \cap \hat{T})$. This implies that $v \notin S_{\mathrm{in}} \cup T_{\mathrm{in}}$ and at least one of S_{out} and T_{out} contains v. Thus, $v \in \mathrm{bd}(\hat{S}) \cup \mathrm{bd}(\hat{T})$.
- $v \in \mathrm{bd}(\hat{S} \cap \hat{T}) \cap \mathrm{bd}(\hat{S} \cup \hat{T})$. This implies that $v \in S_{\mathrm{out}} \cap T_{\mathrm{out}}$ and $v \notin S_{\mathrm{in}} \cup T_{\mathrm{in}}$. Thus, $v \in \mathrm{bd}(\hat{S}) \cap \mathrm{bd}(\hat{T})$.

Hence, the contribution of each vertex to LHS is not less than its contribution to RHS and the statement holds. □

PROPOSITION 2.11. *For any two bisets $\hat{S}$ and $\hat{T}$, $|\mathrm{bd}(\hat{S})| + |\mathrm{bd}(\hat{T})| \ge |\mathrm{bd}(\hat{S} \setminus \hat{T})| + |\mathrm{bd}(\hat{T} \setminus \hat{S})|$.*

PROOF. Consider a vertex v that contributes to RHS. Then it would be one the following cases:

- $v \in \mathrm{bd}(\hat{S} \setminus \hat{T}) \setminus \mathrm{bd}(\hat{T} \setminus \hat{S})$. This implies that $v \in S_{\mathrm{out}} \setminus T_{\mathrm{in}}$. If we have also $v \notin S_{\mathrm{in}}$, then $v \in \mathrm{bd}(\hat{S})$. Otherwise; since $v \notin (S_{\mathrm{in}} \setminus T_{\mathrm{out}})$, we must have $v \in T_{\mathrm{out}}$. Thus, $v \in \mathrm{bd}(\hat{T})$.
- $v \in \mathrm{bd}(\hat{T} \setminus \hat{S}) \setminus \mathrm{bd}(\hat{S} \setminus \hat{T})$. Similar to the above case either $v \in \mathrm{bd}(\hat{S})$ or $v \in \mathrm{bd}(\hat{T})$.
- $v \in \mathrm{bd}(\hat{T} \setminus \hat{S}) \cap \mathrm{bd}(\hat{S} \setminus \hat{T})$. This implies that $v \in T_{\mathrm{out}} \setminus S_{\mathrm{in}}$ and $v \in S_{\mathrm{out}} \setminus T_{\mathrm{in}}$. Thus, $v \in \mathrm{bd}(\hat{S}) \cap \mathrm{bd}(\hat{T})$.

---

[5]Note that these inequalities hold for symmetric submodular set functions and we work with symmetric biset functions throughout the article, since the graphs are undirected.

Hence, the contribution of a vertex to LHS is at least the contribution of the vertex to RHS and the statement holds.   □

Let $G = (V, E)$ be a graph and let $\hat{S}$ be a biset over V. We say an edge e crosses $\hat{S} = (S_{\text{in}}, S_{\text{out}})$ if e has one endpoint S_{in} and the other endpoint in $V \setminus S_{\text{out}}$. For any $F \subseteq E$, we let $\delta_F(\hat{S})$ denote the set of all edges in F that cross $\hat{S}$. We define $\Gamma_F(\hat{S})$ to be the set of all vertices $u \in V \setminus S_{\text{out}}$ for which there exists an edge $uv \in F$ that crosses $\hat{S}$. For a subgraph H, we abuse notation and use $\delta_H(\hat{S})$ to denote $\delta_{E(H)}(\hat{S})$ and $\Gamma_H(\hat{S})$ to denote $\Gamma_{E(H)}(\hat{S})$.

LEMMA 2.12 (LEMMA 3.7 IN [15]). *For any graph $G = (V, E)$ and any subset of edges $F \subseteq E$, $|\delta_F(.)|$ is bisubmodular over $\mathcal{P}$ where $\mathcal{P}$ is any crossing family of bisets over V.*

Definition 2.13 (Feasible Cover). Let $G = (V, E)$ be a graph and let f be an integer-valued function defined on a collection of bisets $\mathcal{P}$ over V. We say that $F \subseteq E$ is a feasible cover of f if $|\delta_F(\hat{S})| \geq f(\hat{S})$ for each $\hat{S} \in \mathcal{P}$. We say that a subgraph $H = (V(H), E(H))$ is a feasible cover of f if $E(H)$ is a feasible cover of f. A subgraph H is a node-minimal cover of f if $H \setminus \{v\}$ is not a feasible cover of f for any $v \in V(H)$.

Definition 2.14 ((Minimal) Violated Biset). Let $G = (V, E)$ be a graph and let f be an integer-valued function defined on a collection of bisets $\mathcal{P}$ over V. For $F \subseteq E$, we say that a biset $\hat{S} \in \mathcal{P}$ is violated with respect to F if $|\delta_F(\hat{S})| < f(\hat{S})$. We say that $\hat{S}$ is a minimal violated biset with respect to F if $\hat{S}$ is violated and there is no violated biset $\hat{T}$ such that $\hat{T} \subset \hat{S}$. These definitions extend to violation with respect to a subgraph H of G.

Definition 2.15 (Non-Overlapping Bisets). Two distinct bisets $\hat{S}$ and $\hat{T}$ are *non-overlapping* iff one of the following holds:

(i) $\hat{S} \subseteq \hat{T}$ or $\hat{T} \subseteq \hat{S}$.
(ii) The sets $S_{\text{out}} \cap T_{\text{in}}$ and $S_{\text{in}} \cap T_{\text{out}}$ are empty.

If the bisets do not satisfy any of the above conditions, then they are *overlapping*.

A useful observation that we will need later is that minimal violated bisets do not overlap with other (not necessarily minimal) violated bisets.

LEMMA 2.16. *Let h be a $\{0, 1\}$-biuncrossable function. Let $\hat{C}$ be a minimal violated biset of h and let $\hat{S}$ be a violated biset of h. Then, $\hat{C}$ and $\hat{S}$ do not overlap. In particular, the inner parts of the minimal violated bisets of h are disjoint.*

PROOF. Since $h(\hat{C}) = h(\hat{S}) = 1$ and h is a biuncrossable function, $h(\hat{C} \cap \hat{S}) = h(\hat{S} \cup \hat{S}) = 1$ or $h(\hat{C} \setminus \hat{S}) = h(\hat{S} \setminus \hat{C}) = 1$. Suppose that the former case holds. Since $\hat{C} \cap \hat{S} \subseteq \hat{C}$, it follows from the minimality of $\hat{C}$ that $\hat{C} \cap \hat{S} = \hat{C}$. Thus, $\hat{C} \subseteq \hat{S}$ and hence $\hat{C}$ and $\hat{S}$ are non-overlapping. Therefore, we may assume that $h(\hat{C} \setminus \hat{S}) = h(\hat{S} \setminus \hat{C}) = 1$. Since $\hat{C} \setminus \hat{S} \subseteq \hat{C}$, it follows from the minimality of $\hat{C}$ that $\hat{C} \setminus \hat{S} = \hat{C}$. Thus, the sets $C_{\text{in}} \cap S_{\text{out}}$ and $C_{\text{out}} \cap S_{\text{in}}$ are empty, and hence $\hat{C}$ and $\hat{S}$ are non-overlapping.

If $\hat{C}$ and $\hat{D}$ are both minimal violated bisets of h, then none of them is a subset of the other one; hence, $C_{\text{in}} \cap D_{\text{out}} = \emptyset$, which implies that $C_{\text{in}} \cap D_{\text{in}}$ is empty as well.   □

Since we are interested in node-weighted problems the subgraphs that arise in our algorithms and analysis are typically node-induced subgraphs. We use the standard terminology of $G[S]$ to denote the subgraph of G induced by a node subset $S \subseteq V(G)$. We use $E[S]$ to denote the set of edges with both end points in S. The graph $G[S] = (S, E[S])$ is the subgraph induced by the vertex set S. We frequently need to consider the graph $(V, E[S])$, and when there is no confusion, we use $G[S]$ to denote this graph as well.

3 ALGORITHM FOR NODE-WEIGHTED ELEM-SNDP

In this section, we formally set up the augmentation framework for node-weighted Elem-SNDP. We point out the specific features of the optimization problem that arises in each phase of the augmentation framework. Our main technical result that provides an $O(1)$-approximation for the augmentation problem of each phase is formally described and analyzed in Section 4.

3.1 Elem-SNDP and Covering Skew-bisupermodular Functions

We set up Elem-SNDP as a special case of covering skew-bisupermodular functions using Menger's theorem for element connectivity. Given an instance of Elem-SNDP over a graph $G = (V, E)$ with requirements specified by r, we extend the requirements to bisets as follows: For each biset $\hat{S}$ defined on V, $r_{\text{elem}}(\hat{S})$ is defined as $\max_{u \in S_{\text{in}}, v \in V \setminus S_{\text{out}}} r(uv)$; in other words, $r_{\text{elem}}(\hat{S})$ is the maximum connectivity requirement over all pair of vertices that are *separated* by $\hat{S}$. Note that we only have connectivity requirement over pairs of reliable nodes. Let $\mathcal{P}_{\text{elem}}$ be the collection of all bisets defined on V whose boundaries only contain *non-reliable* nodes. Then, we define $f_{\text{elem}} : \mathcal{P}_{\text{elem}} \to \mathbb{Z}_+$ as $f_{\text{elem}}(\hat{S}) = r_{\text{elem}}(\hat{S}) - |\text{bd}(\hat{S})|$. The following theorem is not hard to prove and can be found in [10, 15]:

THEOREM 3.1 (MENGER'S THEOREM FOR ELEMENT CONNECTIVITY). *Let $G = (V, E)$ be an undirected graph with V partitioned into reliable nodes R and non-reliable nodes $V \setminus R$. Two distinct nodes $s, t \in R$ are k-element connected iff for each biset $\hat{S} \in \mathcal{P}_{\text{elem}}$ separating s and t, $|\delta(\hat{S})| + |\text{bd}(\hat{S})| \geq k$.*

Applying Menger's theorem, solving node-weighted Elem-SNDP is equivalent to finding a minimum node-weighted (feasible) cover of f_{elem}.

PROPOSITION 3.2. *$\mathcal{P}_{\text{elem}}$ is a crossing bifamily.*

PROOF. $\hat{S} \cap \hat{T} \subseteq \hat{S}$ and $\hat{S} \setminus \hat{T} \subseteq \hat{S}$, which implies that $\text{bd}(\hat{S} \cap \hat{T}) \subseteq \text{bd}(\hat{S})$ and $\text{bd}(\hat{S} \setminus \hat{T}) \subseteq \text{bd}(\hat{S})$; hence, intersection and difference preserve the property that the boundary does not contain any reliable nodes. Recall that $\hat{S} \cup \hat{T} = (S_{\text{in}} \cup T_{\text{in}}, S_{\text{out}} \cup T_{\text{out}})$ and hence $\text{bd}(\hat{S} \cup \hat{T}) = (S_{\text{out}} \cup T_{\text{out}}) \setminus (S_{\text{in}} \cup T_{\text{in}}) \subseteq \text{bd}(\hat{S}) \cup \text{bd}(\hat{T})$; therefore, if both $\hat{S}, \hat{T} \in \mathcal{P}_{\text{elem}}$, then $\hat{S} \cup \hat{T} \in \mathcal{P}_{\text{elem}}$ as well. □

PROPOSITION 3.3. *Let r_{elem} be the requirement function arising from an instance of Elem-SNDP (in other words, r_{elem} is defined on the crossing bifamily $\mathcal{P}_{\text{elem}}$). Then,*

- *$r_{\text{elem}}(\hat{S}) = 0$ for all bisets $\hat{S}$ such that $S_{\text{in}} = \emptyset$ or $S_{\text{out}} = V$.*
- *r_{elem} is skew-bisupermodular and bimaximal.*

PROOF. It is straightforward to see that if $S_{\text{in}} = \emptyset$ or $S_{\text{out}} = V$, then $\hat{S}$ does not separate any pair of terminals and $r_{\text{elem}}(\hat{S}) = 0$.

Fleischer et al. proved that r_{elem} is skew-bisupermodular on $\mathcal{P}_{\text{elem}}$ (see Lemma 3.11 in Reference [15]). Further, we show that r_{elem} is bimaximal on $\mathcal{P}_{\text{elem}}$. Let $\hat{S}, \hat{T} \in \mathcal{P}_{\text{elem}}$ and let (s, t) be a pair of terminals that have the maximum connectivity requirement among all terminal pairs separated by $\hat{S} \cup \hat{T}$, i.e., $r_{\text{elem}}(\hat{S} \cup \hat{T}) = r(s, t)$. Since $s \in S_{\text{in}} \cup T_{\text{in}}$, we have $s \in S_{\text{in}}$ or $s \in T_{\text{in}}$; without loss of generality, assume $s \in S_{\text{in}}$. Since $t \in V \setminus (S_{\text{out}} \cup T_{\text{out}})$, the pair (s, t) is separated by $\hat{S}$ and thus $r_{\text{elem}}(\hat{S} \cup \hat{T}) \leq r_{\text{elem}}(\hat{S}) \leq \max\{r_{\text{elem}}(\hat{S}), r_{\text{elem}}(\hat{T})\}$. □

3.2 Augmentation Framework

Now, we turn our attention to the proof of Theorem 1.1. We alert the reader that, to cover the function f_{elem}, we need to pick a set of *edges*. But, since the weights are (only) on the nodes, we pay for a set of nodes and we can use any of the edges in the graph induced by the selected nodes to cover f_{elem}. More precisely, our goal is to select a minimum-weight subgraph $H = G[X]$ that

covers f_{elem}, where X is a subset of the vertex set of G. We will always assume that X contains all terminals.

Our algorithm for covering f_{elem} uses the augmentation framework introduced by Williamson et al. [37] for edge-weighted EC-SNDP. For a non-negative integer ℓ consider the requirement function r_ℓ where $r_\ell(\hat{S}) = \min\{\ell, r_{\text{elem}}(\hat{S})\}$. Similarly, we define f_ℓ where $f_\ell(\hat{S}) = r_\ell(\hat{S}) - |\text{bd}(\hat{S})|$. The algorithm performs k phases with the following property: At the end of phase ℓ, the algorithm constructs a subgraph H_ℓ that covers f_ℓ. In phase ℓ, the algorithm starts with the subgraph $H_{\ell-1} = (V, E[X_{\ell-1}])$ that covers $f_{\ell-1}$ and adds a new set of vertices to $H_{\ell-1}$ to obtain $H_\ell = (V, E[X_\ell])$. We elaborate on this augmentation process. It is convenient to assume that all vertices in $X_{\ell-1}$ have zero weight, since they have already been paid for. Let $G_\ell = (V, E(G) \setminus E(H_{\ell-1}))$. The goal in phase ℓ is to select a minimum-weight subgraph H of G_ℓ that covers the function h_ℓ, where $h_\ell(\hat{S}) = \max\{0, f_\ell(\hat{S}) - |\delta_{H_{\ell-1}}(\hat{S})|\}$ for each $\hat{S} \in \mathcal{P}_{\text{elem}}$. Note that $h_\ell(\hat{S}) \leq 1$ for all $\hat{S}$. Moreover, it is an uncrossable and bimaximal function and satisfies certain other properties that we will formally specify later.

The phase ℓ augmentation problem is then the following: Given a subgraph $H_{\ell-1}$ that covers $f_{\ell-1}$, find a minimum weight subset of nodes A such that $(V, E_{G_\ell}[X_{\ell-1} \cup A])$ covers h_ℓ.

THEOREM 3.4. *Suppose there is a $\lambda(\ell)$-approximation algorithm for the phase ℓ augmentation problem for each $1 \leq \ell \leq k$. Then there is a $\sum_{\ell=1}^{k} \lambda(\ell)$-approximation for node-weighted Elem-SNDP.*

The preceding theorem is an easy consequence of the augmentation framework and the fact that the optimum cost of any instance of the augmentation problem that arises in phase ℓ is upper bounded by the optimum cost of the solution for the original instance of Elem-SNDP. We will show that $\lambda(\ell) \leq 10$ if G is planar. For proper minor-closed family of graphs, we prove that $\lambda(\ell) = O(1)$ where the constant depends on the family. This leads to the claimed $O(k)$-approximation for node-weighted Elem-SNDP that proves Theorem 1.1.

3.3 Properties of the Function h_ℓ

We now discuss some properties of the function that arises in the augmentation process.

LEMMA 3.5. *The functions r_ℓ, f_ℓ, and $f_\ell - |\delta_{H_{\ell-1}}|$ are skew-bisupermodular on $\mathcal{P}_{\text{elem}}$.*

PROOF. Note that r_ℓ can be defined as the biset function corresponding to the Elem-SNDP instance in which $r_\ell(s, t) = \min\{\ell, r(s, t)\}$. By Proposition 3.3, r_ℓ is a skew-bisupermodular function. Since $|\text{bd}(\hat{S})|$ is a bisubmodular function (Proposition 2.9), by Proposition 2.8, f_ℓ is skew-bisupermodular as well. Moreover, using the fact that f_ℓ is skew-bisupermodular and $|\delta_{H_{\ell-1}}(.)|$ is bisubmodular, by Proposition 2.8, $f_\ell - |\delta_{H_{\ell-1}}|$ is skew-bisupermodular as well. □

For each biset $\hat{S} \in \mathcal{P}_{\text{elem}}$, let $h'_\ell(\hat{S}) = f_\ell(\hat{S}) - |\delta_{H_{\ell-1}}(\hat{S})|$. From the preceding lemma h'_ℓ is skew-bisupermodular and moreover for each $\hat{S} \in \mathcal{P}_{\text{elem}}$, $h'_\ell(\hat{S}) \leq 1$. Note that $h_\ell(\hat{S}) = \max\{0, h'_\ell(\hat{S})\}$. We claim that h_ℓ is bi-uncrossable. To see this, suppose $h_\ell(\hat{S}) = 1$ and $h_\ell(\hat{T}) = 1$, then by skew-supermodularity of h'_ℓ, we have $h'_\ell(\hat{S} \cup \hat{T}) + h'_\ell(\hat{S} \cap \hat{T}) \geq 2$ or $h'_\ell(\hat{S} \setminus \hat{T}) + h'_\ell(\hat{T} \setminus \hat{S}) \geq 2$; this is possible only if $h_\ell(\hat{S} \cup \hat{T}) + h_\ell(\hat{S} \cap \hat{T}) \geq 2$ or $h_\ell(\hat{S} \setminus \hat{T}) + h_\ell(\hat{T} \setminus \hat{S}) \geq 2$, because both h_ℓ and h'_ℓ are at most 1 on any biset.

PROPOSITION 3.6. *Consider an integer $\ell \leq k$. Then $h_\ell(\hat{S}) = 1$ iff $r_{\text{elem}}(\hat{S}) \geq \ell$ and $|\text{bd}(\hat{S})| + |\delta_{H_{\ell-1}}(\hat{S})| = \ell - 1$.*

PROOF. If $r_{\text{elem}}(\hat{S}) \geq \ell$ and $|\text{bd}(\hat{S})| + |\delta_{H_{\ell-1}}(\hat{S})| = \ell - 1$, then by definition, $h_\ell(\hat{S}) = \max\{0, \ell - \text{bd}(\hat{S}) - |\delta_{H_{\ell-1}}(\hat{S})|\} = 1$.

We now consider the other direction. Suppose that $h_\ell(\hat{S}) = 1$. This implies that $f_\ell(\hat{S}) - |\delta_{H_{\ell-1}}(\hat{S})| = 1$. Since $H_{\ell-1}$ covers $f_{\ell-1}$, we have $|\delta_{H_{\ell-1}}(\hat{S})| \geq f_{\ell-1}(\hat{S})$. Thus, $f_\ell(\hat{S}) \geq f_{\ell-1}(\hat{S}) + 1$ and hence $r_\ell(\hat{S}) \geq r_{\ell-1}(\hat{S}) + 1$. It follows that $r_{\text{elem}}(\hat{S}) \geq \ell$ and $|\text{bd}(\hat{S})| + |\delta_{H_{\ell-1}}(\hat{S})| \leq r_\ell(\hat{S}) - 1 = \ell - 1$. Moreover, since $|\delta_{H_{\ell-1}}(\hat{S})| + |\text{bd}(\hat{S})| \geq r_{\ell-1}(\hat{S}) = \ell - 1$, $|\delta_{H_{\ell-1}}(\hat{S})| + |\text{bd}(\hat{S})| = \ell - 1$. □

Recall that $H_{\ell-1}$ covers $f_{\ell-1}$ and $G_\ell = G(V, E(G) \setminus E(H_{\ell-1}))$. Further $H_{\ell-1} = (V, E[X_{\ell-1}])$ where $X_{\ell-1}$ is the set of nodes paid for in the first $\ell - 1$ phases.

LEMMA 3.7. *For any $X \supset X_{i-1}$ let $H_X = (V, E_{G_\ell}[X])$ be a subgraph of G_ℓ. Suppose $\hat{C} \in \mathcal{P}_{\text{elem}}$ is a violated biset of H_X with respect to h_ℓ. Then $\text{bd}(\hat{C}) \subseteq X$.*

PROOF. Suppose for the sake of contradiction there is a violated biset $\hat{C}$ and a vertex $u \in \text{bd}(\hat{C})$ such that $u \notin X$. Consider the biset $\hat{C}_1 = (C, \text{bd}(\hat{C}) \setminus \{u\})$. Since u is not a terminal, $r_\ell(\hat{C}_1) = r_\ell(\hat{C}) = \ell$. Since $h_\ell(\hat{C}) = 1$, we have $r_\ell(\hat{C}) = \ell$ and $|\text{bd}(\hat{C})| + |\delta_{H_{\ell-1}}(\hat{C})| = \ell - 1$. In the graph $H_{\ell-1}$, the vertex u has no edges incident to it, since $H_{\ell-1} = (V, E[X_{\ell-1}])$ and $u \notin X_{\ell-1}$. Therefore, $\delta_{H_{\ell-1}}(\hat{C}_1) = \delta_{H_{\ell-1}}(\hat{C})$. Since $|\text{bd}(\hat{C}_1)| = |\text{bd}(\hat{C})| - 1$, we have $|\text{bd}(\hat{C}_1)| + |\delta_{H_{\ell-1}}(\hat{C}_1)| = \ell - 2$, which implies that $H_{\ell-1}$ is not a feasible cover for $r_{\ell-1}$, a contradiction. □

LEMMA 3.8. *For any $X \supset X_{i-1}$ let $H_X = (V, E_{G_\ell}[X])$ be a subgraph of G_ℓ. Suppose $\hat{C} \in \mathcal{P}_{\text{elem}}$ is a* minimal *violated biset of H_X with respect to h_ℓ. Then the following properties hold:*

- $C_{\text{out}} \subseteq X$.
- $G[C_{\text{in}}]$ *is a connected subgraph of G.*

PROOF. For ease of notation, we let H denote the subgraph H_X. Note that since $\hat{C}$ is a violated biset in H, we have $h_\ell(\hat{C}) = 1$ and $|\delta_H(\hat{C})| = 0$. From Proposition 3.6, since $h_\ell(\hat{C}) = 1$, $r(\hat{C}) = \ell$ and $|\text{bd}(\hat{C})| + |\delta_{H_{\ell-1}}(\hat{C})| = \ell - 1$. Suppose there is a vertex $u \in C_{\text{out}}$ such that $u \notin X$. By Lemma 3.7, $u \in C_{\text{in}}$. First, u is not a terminal, since all terminals are in X_{i-1} (and hence in X). Second, u is an isolated vertex in H, since the only edges in H are between nodes in X. Consider the biset $\hat{C}_1 = (C_{\text{in}} - u, C_{\text{out}} - u)$ obtained from $\hat{C}$ by removing u from both inner and outer parts of $\hat{C}$. Since u is not a terminal, we have $r(\hat{C}) = r(\hat{C}_1)$. And, since u is isolated in H, we have $\delta_H(\hat{C}_1) = \delta_H(\hat{C})$, since we moved u out of C', $\text{bd}(\hat{C}_1) \subseteq \text{bd}(\hat{C})$. These facts imply that $h_\ell(\hat{C}_1) = 1$ and $\hat{C}_1$ is a violated biset in H. This contradicts minimality of $\hat{C}$. Therefore, $C' \subseteq X$.

Second, we now prove that $G[C_{\text{in}}]$ is connected. For sake of contradiction, suppose it is not. Let $\overline{C}_1, \overline{C}_2$ be two non-empty sets that partition C_{in} such that there is no edge between $\overline{C}_1$ and $\overline{C}_2$ in G; such a partition exists if $G[C_{\text{in}}]$ is not connected. Note that $E_H(\overline{C}_1, \overline{C}_2) = \emptyset$, since H is a subgraph of G. Define $\hat{C}_1 = (C_1, C_1 \cup \text{bd}(\hat{C}))$ and $\hat{C}_2 = (C_2, C_2 \cup \text{bd}(\hat{C}))$. Since r is bimaximal (Proposition 3.6), $r(\hat{C}) \leq \max\{r(\hat{C}_1), r(\hat{C}_2)\}$. Thus, without loss of generality, we can assume that $r(\hat{C}_1) \geq r(\hat{C}) \geq \ell$. Since $E_H(C_1, C_2) = \emptyset$, we have $\delta_H(\hat{C}_1) \subseteq \delta_H(\hat{C})$. Since $\text{bd}(\hat{C}_1) = \text{bd}(\hat{C})$ and $\hat{C}$ was a violated biset, it follows that $\hat{C}_1$ is also a violated biset with respect to h_ℓ in H. This contradicts the minimality of $\hat{C}$. □

The collection of minimal violated bisets in H_X with respect to h_ℓ are disjoint (due to the bi-uncrossability), and they can computed in polynomial time via Menger's theorem for element-connectivity and standard maxflow algorithms. We refer the reader to [15, 25].

4 APPROXIMATION ALGORITHM FOR THE AUGMENTATION PROBLEM

In this section, we design an $O(1)$-approximation algorithm for minimum node-weighted cover for the augmentation problem that needs to be solved in each of the k phases of the augmentation framework that was described in the preceding section. We recall the problem arises in the phase

ℓ of the augmentation framework. We are given $X_{\ell-1} \subseteq V$ such that the graph $H_{\ell-1} = (V, E[X_{\ell-1}])$ is a feasible cover for $f_{\ell-1}$. The goal is to find a minimum-weight subset of nodes $A \subseteq V \setminus X_{\ell-1}$ such that $H_\ell = (V, E[X_{\ell-1} \cup A])$ covers f_ℓ. This is recast as the problem of covering the $\{0, 1\}$-biuncrossable function h_ℓ in the graph $G_\ell = (V, E(G) \setminus E(H_{\ell-1}))$.

In the edge-weighted case, one can obtain a 2-approximation for covering a $\{0, 1\}$-biuncrossable function in a general undirected graph provided the function has some reasonable computational properties such as the ability to efficiently find the minimal violated sets with respect to any subset of the given edges of a graph. In particular, a natural LP relaxation has an integrality gap of at most 2. However, in the node-weighted setting, a natural LP relaxation that we will discuss shortly has an unbounded integrality gap. However, the function h_ℓ that arises in the augmentation framework for Elem-SNDP has additional properties that allow us to prove a constant factor approximation in planar graphs. One can also prove an $O(\log n)$-approximation in general graphs and this cannot be improved, since node-weighted Steiner tree is a special case that generalizes the Set Cover problem.

In this section, we prove the following theorem:

THEOREM 4.1. *Let $G = (V, E)$ be a node-weighted graph from a proper minor-closed family of graphs $\mathcal{G}$. Let h_ℓ be $\{0, 1\}$-biuncrossable function that arises in phase ℓ of the augmentation framework for an instance of Elem-SNDP defined over the graph G. There exists an $O(1)$-approximation algorithm for the problem of finding a minimum-weight node subset to cover h_ℓ.*

We remark that our result applies to a class of $\{0, 1\}$-biuncrossable functions that is more general than the class of functions that arise from Elem-SNDP. Characterizing the precise class for which the algorithm applies is not quite as clean as we would like and, hence, we do not attempt to do so.

Our algorithm is a primal-dual algorithm modeled after the well-studied algorithm for the edge-weighted case [1, 19]. The adaptation of the primal-dual algorithm to the node-weighted case in planar graphs was done in [13] but were concerned with Steiner forest and $\{0, 1\}$ proper functions while our setting is more general.

4.1 A Primal-dual Algorithm

Instead of focusing on the specific setting of covering the restricted class of functions that arise in Elem-SNDP, we will work in an abstract framework where we have a general $\{0, 1\}$-biuncrossable function h defined over a crossing bifamily $\mathcal{P}$ in a node-weighted graph $G = (V, E)$. The goal is to find a minimum weight subset $X \subseteq V$ such that the subgraph $H = (V, E[X])$ covers h, that is, $|\delta_H(\hat{S})| \geq h(\hat{S})$ for each $\hat{S} \in \mathcal{P}$.

LP relaxation. We consider a natural LP-relaxation of the problem and its dual, which are shown in Figure 1. There is a variable $x(v)$ that in the integer programming formulation indicates whether v is chosen and in the LP relaxation $x(v)$ is relaxed to be in the interval $[0, 1]$. Consider a biset $\hat{S} \in \mathcal{P}$ such that $h(\hat{S}) = 1$. Then any subgraph H of G that covers h needs to contain an edge $e \in \delta_G(\hat{S})$, which implies that there is an endpoint v of e such that $v \in \Gamma(\hat{S})$; therefore, at least one vertex in $\Gamma(\hat{S})$ needs to be included in any feasible cover of h. This justifies the constraint in the LP relaxation. Note that we omitted the constraint $x(v) \leq 1$ from the primal, since it is redundant.

LEMMA 4.2. *The Primal-LP is a valid relaxation of the problem of covering 0-1 biset functions on node-weighted graphs.*

Integrality gap example. Before we describe the primal-dual algorithm, we describe a simple example to demonstrate that the integrality gap of the LP is unbounded for general biuncrossable functions. Let $G = (V, E)$ be a complete graph on $n \geq 3$ nodes. Consider the case when $\mathcal{P}$ is the

Primal-LP ▷ Input: $(G, \mathcal{P}, h)$

$$\min \sum_{v \in V} w(v) x(v)$$

$$\text{s.t.} \sum_{v \in \Gamma(\hat{S})} x(v) \geq h(\hat{S}) \quad \forall \hat{S} \in \mathcal{P}$$

$$x(v) \geq 0 \quad \forall v \in V$$

Dual-LP ▷ Input: $(G, \mathcal{P}, h)$

$$\max \sum_{\hat{S} \in \mathcal{P}} h(\hat{S}) y(\hat{S})$$

$$\text{s.t.} \sum_{\hat{S}: v \in \Gamma(\hat{S})} y(\hat{S}) \leq w(v) \quad \forall v \in V$$

$$y(\hat{S}) \geq 0 \quad \forall \hat{S} \in \mathcal{P}$$

Fig. 1. LP-relaxation of the optimization problem of covering biset functions on node-weighted graphs and its dual program.

set of all bisets over V and h is the function such that $h(\hat{S}) = 1$ for a biset $\hat{S}$ where $S_{\text{in}} = \{v_1\}$ and $S_{\text{out}} = \{v_1\}$; $h(\hat{T}) = 0$ for all other bisets. It is easy to see that h is biuncrossable. Let $w(v_1) = 1$ and $w(v_i) = 0$ for all $i \geq 2$. The only way to cover h is to pick v_1 and one other node and hence the optimum solution has weight 1. However, the LP relaxation is forced to only pick a neighbor of v_1 and pays 0. This is true even if the variables are required to be integer and in fact the integer solution is not necessarily even feasible for the original problem. The technical issue here is that there is no notion of "terminals" when working with a general biuncrossable function. However, when working with Elem-SNDP the terminals are always included in a solution and can be assumed to have weight 0. As the algorithm proceeds, the newly added vertices can be treated as terminals and the connectivity properties satisfied by h_ℓ help in this regard. This will become clearer in the analysis.

Primal-dual algorithm. We now describe the primal-dual algorithm. It is inspired by the one in [13], which is an adaptation to the node-weighted setting of the standard primal-dual algorithm for the edge-weighted case [1, 19]. The algorithm selects a subset of vertices P such that the graph $(V, E[P])$ covers h. However, we need to include, at the start of the algorithm, a prespecified subset of vertices whose weight will not be counted in analyzing the performance of the algorithm; alternatively, we can assume that these nodes have weight 0. In [13] the prespecified subset is the set of terminals. In the augmentation framework this is the set of all the vertices that have been selected in the previous phases (for phase 1 this is the set of terminals).

The algorithm has two high-level stages. In the first stage it starts with P_0, the set of all zero weight vertices, and iteratively adds vertices to P_0 as long as there are violated bisets. This state is guided by maintaining a dual feasible solution $\mathbf{y}$ that is implicitly initialized to zero. The iterations proceed as follows: Consider iteration i and let P_{i-1} be the set of all nodes selected in the first $i - 1$ iterations. Let C_i be the collection of all minimal violated bisets of h with respect to graph $G_i = (V, E[P_{i-1}])$. C_i can be computed in polynomial-time, and any two bisets in this family do not cross by the biuncrossability property of h. The algorithm assumes access to a procedure VIOLATEDBISETS that outputs the minimal violated bisets with respect to a given subgraph. The first stage of the algorithm stops when C_i is empty. Otherwise, in iteration i it increases the dual variables $\{y(\hat{C})\}_{\hat{C} \in C_i}$ uniformly until a dual constraint for a vertex v becomes tight, that is, we have $\sum_{\hat{S}: v \in \Gamma(\hat{S})} y(\hat{S}) = w(v)$. If the dual constraint corresponding to v becomes tight, then we add v to P_{i-1} to obtain P_i and move to iteration $i + 1$. If several vertices become tight at the same time, then we pick one of them arbitrarily.

The second stage of the algorithm is a *reverse-delete* step. Let P be the set of vertices selected by the primal-dual algorithm. We select a subset Q of P as follows: We start with $Q = P$. We order the vertices of Q in the reverse of the order in which they were selected by the primal-dual algorithm. Let v be the current vertex. If $(V, E[Q \setminus \{v\}])$ is still a feasible cover for h, then we remove v from Q. The algorithm outputs the vertices that remain in Q.

COVER(G, h):
   ▷ **y** denotes the variables of Dual-LP and initialized to zero
   y ← 0
   $P_0 \leftarrow \{v \mid w(v) = 0\}, P \leftarrow P_0$
   $i \leftarrow 1$
   ▷ VIOLATEDBISETS(G, h, P) returns all minimal violated bisets of h w.r.t $G[P]$
   $C_i \leftarrow$ VIOLATEDBISETS(G, h, P_{i-1})
   while $C_i \neq \emptyset$
       Increase $y(\hat{S})$ uniformly for all $\hat{S} \in C_i$ until a constraint of
       Dual-LP becomes tight (for $v, \sum_{\hat{S}:v \in \Gamma(\hat{S})} y(\hat{S}) = w(v)$)
       $P_i \leftarrow P_{i-1} \cup \{v\}, P \leftarrow P_i$
       $i \leftarrow i + 1$
       $C_i \leftarrow$ VIOLATEDBISETS(G, h, P_{i-1})

   ▷ *reverse-delete* step:
   $Q \leftarrow P$
   for each $v \in Q$ in the reverse of the order in which the **while** loop added vertices
       if VIOLATEDBISETS($G, h, Q \setminus \{v\}$) = $\emptyset$
          $Q \leftarrow Q \setminus \{v\}$

Fig. 2. A primal-dual algorithm for covering restricted $\{0, 1\}$-biuncrossable functions.

A formal description of the described primal-dual algorithm is given in Figure 2. As we discussed earlier, the algorithm is not guaranteed to output a feasible solution for an arbitrary $\{0, 1\}$-biuncrossable function. However, we argue below that it returns a feasible solution for the functions that arise in the augmentation framework. We will assume that the algorithm is run on graph G_ℓ with function h_ℓ and $P_0 = X_{\ell-1}$.

PROPOSITION 4.3. *At the start of iteration i of the first while loop, we have $P_{i-1} \cap \Gamma(\hat{C}) = \emptyset$ for every $\hat{C} \in C_i$.*

PROOF. Each biset $\hat{C} \in C_i$ is a minimal violated biset with respect to the graph $G_\ell[P_{i-1}]$. From Lemma 3.8 it follows that $C_{\text{out}} \subseteq P_{i-1}$ and hence $\Gamma(\hat{C})$ can only contain vertices in $V \setminus P_{i-1}$. □

The lemma below shows that the algorithm maintains dual feasibility with respect to the primal solution P.

LEMMA 4.4. *The dual solution y constructed by the primal-dual algorithm satisfies the primal complementary slackness conditions. More precisely, for each $v \in P$, $\sum_{\hat{S}:v \in \Gamma(\hat{S})} y(\hat{S}) = w(v)$.*

PROOF. We prove the lemma by induction on the number of iterations of the first while loop. Initially, y is zero and P_0 consists of all zero-weight vertices. Thus, the complementary slackness conditions are satisfied at the beginning of the algorithm. Now consider iteration $i > 0$. From Proposition 4.3, no vertex in P_{i-1} is adjacent, in the graph G_ℓ, to any biset in C_i. Thus, increasing the dual variables corresponding to bisets of C_i do not violate the tightness of vertices in P_{i-1}. And the only vertex added to P_i in iteration i is the one that becomes tight with respect to the dual increase in iteration i. Thus, at the end of iteration i, the required condition holds for P_i. □

LEMMA 4.5. *The primal-dual algorithm returns a feasible cover for h_ℓ in G_ℓ when all vertices of $X_{\ell-1}$ are included in P_0.*

Proof. Proposition 4.3 and Lemma 4.4 show that the algorithm inductively maintains the property that C_i is the collection of minimal violated bisets with respect to the graph $G_\ell[P_{i-1}]$; hence, when the first while loop terminates, we have the property that there are no minimal violated bisets with respect to $G_\ell[P]$, and thus $G_\ell[P]$ is a feasible cover. The reverse-delete step explicitly ensures that $G_\ell[Q]$ is a feasible cover. □

4.2 Analysis of the Approximation Ratio

We now analyze the approximation ratio. We will assume that the function h comes from an augmentation problem and that the algorithm is well-defined and returns a feasible cover. The basic lemma that underlies the primal-dual analysis is similar to that of the edge-weighted case and relies on the uniform-growth property of the dual variables.

Lemma 4.6. *Let Q be the set of vertices output by the primal-dual algorithm. Suppose that there exists a fixed value γ such that, for each iteration i of the primal-dual algorithm, $\sum_{\hat{C} \in C_i} |Q \cap \Gamma(\hat{C})| \leq \gamma |C_i|$. Then $w(Q)$ is at most γ times the value of an optimal solution of $\text{Primal-LP}(G, \mathcal{P}, h)$.*

The content of the preceding lemma is the following: Consider the minimal violated bisets in iteration i, C_i. Let $Q_i = Q \setminus P_{i-1}$. From the reverse-delete step, we can see that Q_i forms a *node-minimal* set that together with P_{i-1} covers h. We are interested in γ, the "average degree"[6] of the bisets in C_i, with respect to nodes in Q_i. In general graphs, γ can be $\Omega(n)$ in the worst case and we will not be able to prove any reasonable guarantee on the performance of the primal-dual algorithm. However, planar graphs and more generally the graphs from minor-closed families are sparse. Thus, we can bound the average degree if we upper bound the number of nodes in Q_i that are neighbors of a biset in C_i. The following proof follows a standard template in the context of primal-dual analysis, but we give it here for the sake of completeness:

Proof of Lemma 4.6. By Lemma 4.4, y satisfies the primal complementary slackness conditions. Therefore, we have

$$\sum_{v \in Q} w(v) = \sum_{v \in Q} \sum_{\hat{S} \in \mathcal{P}: v \in \Gamma(\hat{S})} y(\hat{S}) = \sum_{\hat{S} \in \mathcal{P}} y(\hat{S}) |Q \cap \Gamma(\hat{S})|.$$

Note that, if we can show that $\sum_{\hat{S} \in \mathcal{P}} y(\hat{S}) |Q \cap \Gamma(\hat{S})| \leq \gamma \sum_{\hat{S} \in \mathcal{P}} y(\hat{S}) h(\hat{S})$, then it will follow that we have a γ-approximation: Since y is feasible, $\sum_{\hat{S} \in \mathcal{P}} y(\hat{S}) h(\hat{S})$ is a lower bound on the fractional optimum, which in turn is a lower bound on the integral optimum.

We show by induction on the number of iterations of the primal-dual algorithm that

$$\sum_{\hat{S} \in \mathcal{P}} y(\hat{S}) |Q \cap \Gamma(\hat{S})| \leq \gamma \sum_{\hat{S} \in \mathcal{P}} y(\hat{S}) h(\hat{S}).$$

Note that $y(\hat{S}) > 0$ only if $h(\hat{S}) = 1$. Therefore, $\sum_{\hat{S} \in \mathcal{P}} y(\hat{S}) h(\hat{S}) = \sum_{\hat{S} \in C} y(\hat{S})$ where C is the collection of violated bisets with respect to G. In the primal-dual algorithm, we only increase the dual values of violated bistes and thus it suffices to prove that

$$\sum_{\hat{S} \in C} y(\hat{S}) |Q \cap \Gamma(\hat{S})| \leq \gamma \sum_{\hat{S} \in C} y(\hat{S}).$$

Initially, all dual variables $y(\hat{S})$ are zero and therefore the inequality holds. Now consider iteration i of the primal-dual algorithm. Recall that P_{i-1} is the set of all vertices selected in the first $i - 1$

---

[6]Here, we are abusing the term slightly and we refer to the ratio $\sum_{\hat{C} \in C_i} |Q_i \cap \Gamma(\hat{C})| / |C_i|$ as the average degree of the bisets in C_i.

iterations of the primal-dual algorithm, and C_i is the set of all minimal violated bisets with respect to $G_\ell[P_{i-1}]$.

Let ϵ denote the amount by which we increased $y(\hat{S})$ for $\hat{S} \in C_i$ in iteration i. The left-hand side increases by $\sum_{\hat{C} \in C_i} \epsilon |Q \cap \Gamma(\hat{C})|$, and the right-hand side increases by $\gamma \epsilon |C_i|$. Therefore, it suffices to show that

$$\sum_{\hat{C} \in C_i} |Q \cap \Gamma(\hat{C})| \le \gamma |C_i|.$$

Recall that $Q_i = Q \setminus P_{i-1}$. By Proposition 4.3, for each $\hat{C} \in C_i$, $\Gamma(\hat{C}) \cap P_{i-1}$ is empty and thus $\Gamma(\hat{C}) \cap Q = \Gamma(\hat{C}) \cap Q_i$. Therefore, we can rewrite the inequality above as:

$$\sum_{\hat{C} \in C_i} |Q_i \cap \Gamma(\hat{C})| \le \gamma |C_i|,$$

which holds by the assumption in the statement of the lemma.   □

The key technical contribution of the article is to bound γ, and it is captured by the following theorem whose proof is in Section 4.3:

THEOREM 4.7. *Consider phase ℓ of the augmentation algorithm. For $X \supset X_{i-1}$ let $H_X = (V, E[X])$ be a subgraph of G_ℓ. Let C be the collection of minimal violated bisets of h_ℓ with respect to H_X. Suppose $Q \subseteq V \setminus X$ is a node-minimal set such that $G_\ell[X \cup Q]$ is a feasible cover for h_ℓ. Then $|Q \cap (\bigcup_{\hat{C} \in C} \Gamma(\hat{C}))| \le 4|C|$.*

Exploiting the sparsity of planar graphs and more generally minor-closed families of graphs together with the preceding theorem, we obtain the following lemma:

LEMMA 4.8. *Consider an instance of Elem-SNDP over a graph G that belongs to a proper minor-closed family of graphs $\mathcal{G}$. Suppose we run the primal-dual algorithm in phase ℓ to cover $h = h_\ell$. Let $Q_i = Q \setminus P_{i-1}$. Then, $\sum_{\hat{C} \in C_i} |Q_i \cap \Gamma(\hat{C})| \le c_{\mathcal{G}} |C_i|$, where $c_{\mathcal{G}}$ is a constant that depends only on the family $\mathcal{G}$. In particular, if G is a planar graph, then $\sum_{\hat{C} \in C_i} |Q_i \cap \Gamma(\hat{C})| \le 10|C_i|$.*

PROOF. Let $N_i = Q_i \cap (\bigcup_{\hat{C} \in C_i} \Gamma(\hat{C}))$. By Theorem 4.7, $|N_i| \le 4|C_i|$. Let $C_i = \{\hat{C}_1 = (C_1, C_1'), \hat{C}_2 = (C_2, C_2'), \ldots, \hat{C}_r = (C_r, C_r')\}$ where for each $i \in [r]$, C_i, C_i' are, respectively, the inner and outer part of the biset $\hat{C}_i$. Since C_i is the collection of minimal violated of h_ℓ and h_ℓ is a biuncrossable function, the collection of inner sets $C_1, C_2, \ldots, C_r$ are pairwise disjoint. We also have the property that $N_i \cap C_j = \emptyset$ for $1 \le j \le r$. Further, from Lemma 3.8, $G[C_j]$ is connected for $1 \le j \le r$. Next, we construct a minor K of G as follows: Let $V' = (\bigcup_{\hat{C} \in C_i} C) \cup N_i$. Note that we do not include the boundary vertices of the bisets of C_i in V'. We start with $K = G[V']$. For each biset $\hat{C}_j \in C_i$, we shrink the set C_j to a single vertex v_j. We also remove parallel edges to get a simple graph. The resulting graph is indeed a minor of G, since (i) $G[C_j]$ is connected for each j, and (ii) $C_1, C_2, \ldots, C_r, N_i$ are pairwise disjoint. The total number of nodes in K is $|N_i| + |C_i| \le 5|C_i|$. In K, we also remove the edges with both endpoints in N_i, which results in a bipartite graph with N_i on one side, and the vertices $v_1, v_2, \ldots, v_r$ corresponding to $C_1, C_2, \ldots, C_r$ on the other side. Note that K is still a minor of G.

Recall that we have $Q_i \cap \Gamma(\hat{C}) \subseteq N_i$ for each $\hat{C} \in C_i$. Therefore, $\sum_{\hat{C} \in C_i} |Q_i \cap \Gamma(\hat{C})|$ is equal to the number of edges in the bipartite graph K. Since K is from a minor-closed family $\mathcal{G}$, from [28] it follows that there is a constant $c_{\mathcal{G}}'$ that depends only on the family such that $|E(K)| \le c_{\mathcal{G}}' |V(K)| \le 5c_{\mathcal{G}}' |C_i|$. Suppose G is a planar graph. Then K is a bipartite planar graph and in this case it is well-known that $|E(K)| \le 2|V(K)| \le 10|C_i|$.   □

4.3 Proof of Theorem 4.7

In this section, we prove Theorem 4.7 using a counting argument that is a generalization of the counting argument of [8]. We use H in place of H_X to simplify the notation. We use K to denote the graph $(V(G), E(G) \setminus E(H))$. Consider the biset function h' where $h'(\hat{S}) = 1$ iff $h_\ell(\hat{S}) = 1$ and $\delta_H(\hat{S}) = \emptyset$. By Proposition 2.8 and Lemma 2.12, h' is a $\{0, 1\}$-biuncrossable function. Note that h' is the residual function of h_ℓ in the graph H. Let $Q' = Q \cup X$. Recall that Q a is node-minimal set such that $G_\ell[Q']$ covers h_ℓ. Equivalently this means that Q is a node-minimal set such that $K[Q \cup X]$ covers h'.

The main idea in the proof is to pick a subset M of the edges of $K[Q']$ such that M is an *edge-minimal* feasible cover for h'. An edge-minimal set allows us to use an approach that was introduced by Williamson et al. [37] for the edge-weighted SNDP problem. More precisely, for each edge $e \in M$, we can pick a *witness biset* that is a violated biset of h' such that e is the only edge of M that is leaving the biset. Moreover, we can pick a laminar family of witness bisets for all edges in M that allows us to upper bound the number of edges in M incident to the components of C in terms of $|C|$.

Since $K[Q']$ is a node-minimal cover of h' and not an edge-minimal cover, it is possible that there is a vertex $u \in Q$ connected to a component of C in $K[Q']$ but none of the edges connecting u to components of C are in M. Thus, we cannot use the family of witness bisets of an edge-minimal cover of h' to bound the number of these vertices. We address this issue by counting these vertices separately using a witness family of bisets for a different set of *non-redundant* edges.

We refer to the vertices in $Q \cap \left(\bigcup_{\hat{C} \in C} \Gamma_G(\hat{C}) \right)$ as **critical** vertices; these are the vertices of Q that are adjacent to at least one biset in C and the goal is to show that there are at most $4|C|$ of them. We refer to the edges in $\cup_{\hat{C} \in C} \delta_K(\hat{C})$ as **red edges** and all other edges of K as **blue edges**. Every critical vertex is incident to at least one red edge. We define two subsets of edges F and F' below.

We start with $F = E(K)$ and we remove some of the edges as follows: We consider the *blue edges* in an arbitrary order. Let e be the current edge. If $F \setminus \{e\}$ is a feasible solution for h', then we remove e from F. This procedure gives us a set of edges in which each blue edge is necessary, in the sense that removing any blue edge from F will make it an infeasible cover for h'. As we will see shortly, we can use the blue edges in F to upper bound the number of critical vertices that are incident to at least one blue edge of F. We refer to critical vertices that are incident to a blue edge of F as **regular** vertices, and we refer to all other critical vertices as **special** vertices.

To count the special vertices, we pick a subset F' of F as follows: We start with $F' = F$ and consider the *red edges* of F' in some arbitrary order. Let e be the current edge. If $F' \setminus \{e\}$ is a feasible cover of h', then we remove e from F'. We can use the red edges in F' to upper bound the remaining critical vertices. Since Q is a node-minimal cover for h', each special vertex is incident to at least one red edge of F'.

We consider the regular and special vertices separately. Theorem 4.7 follows from the following lemmas:

LEMMA 4.9. *The number of regular vertices is at most $2|C|$.*

LEMMA 4.10. *The number of special vertices is at most $2|C|$.*

First, we formally define the notion of *witness bisets* and discuss the existence of a family of *non-overlapping* bisets that is a key part in the counting argument.

Definition 4.11 (Witness Biset). Let $G = (V, E)$ be an input graph and let h be a $\{0, 1\}$-biuncrossable function defined on a crossing bifamily $\mathcal{P} \subseteq 2^V \times 2^V$. Let $F \subseteq E$ be a feasible cover of h. Then, $\hat{S}_e$ is an F-witness biset of $e \in F$ iff $h(\hat{S}_e) = 1$ and $\delta_F(\hat{S}_e) = \{e\}$.

Definition 4.12 (Laminar Bifamily). A family of bisets $\mathcal{F}$ is laminar iff for any $\hat{S}, \hat{T} \in \mathcal{F}$, $\hat{S}$ and $\hat{T}$ are non-overlapping.

Given a feasible cover F for a requirement function h, we say that an edge $e \in F$ is *non-redundant* if $F \setminus \{e\}$ is not a cover. A set $M \subseteq F$ is non-redundant if each edge $e \in M$ is non-redundant. The following lemma is known from past work and we provide a proof for the sake of completeness:

LEMMA 4.13. *Let F be a feasible cover of a 0-1 bi-uncrossable function h. Let $M \subseteq F$ be a set of non-redundant edges with respect to F. There exists a laminar family of bisets $\mathcal{L} = \{\hat{S}_e \mid e \in M\}$ such that $\hat{S}_e$ is an F-witness biset for e.*

To prove Lemma 4.13, our first observation is that, if M is a set of non-redundant edges, then we can pick a witness biset for each edge of M.

LEMMA 4.14 (LEMMA 4.1 IN [15]). *Let $\hat{S}_{e_1}, \hat{S}_{e_2}$ be F-witness bisets for e_1 and e_2, respectively. Then one of the following holds:*

   (i) *The bisets $\hat{S}_{e_1} \cap \hat{S}_{e_2}$ and $\hat{S}_{e_1} \cup \hat{S}_{e_2}$ are F-witness bisets for distinct edges in $\{e_1, e_2\}$.*
   (ii) *The bisets $\hat{S}_{e_1} \setminus \hat{S}_{e_2}$ and $\hat{S}_{e_2} \setminus \hat{S}_{e_1}$ are F-witness bisets for distinct edges in $\{e_1, e_2\}$.*

Using the following lemma, we can show that, if we replace two overlapping witness bisets with the witness bisets guaranteed by Lemma 4.14, the number of pairs of overlapping bisets decreases and, thus, we are making progress towards a laminar family of witness bisets:

LEMMA 4.15 (LEMMA 4.3 IN [15]). *Let $\hat{S}_1$, $\hat{S}_2$ be two overlapping bisets and $\hat{T}$ be an arbitrary biset. Then the number of pairs of $\{(\hat{S}_1, \hat{S}_2), (\hat{S}_1, \hat{T}), (\hat{S}_2, \hat{T})\}$ that overlap is at least the number of pairs of $\{(\hat{S}_1 \cap \hat{S}_2, \hat{S}_1 \cup \hat{S}_2), (\hat{S}_1 \cap \hat{S}_2, \hat{T}), (\hat{S}_1 \cup \hat{S}_2, \hat{T})\}$ that overlap. Similarly, the number of pairs of $\{(\hat{S}_1, \hat{S}_2), (\hat{S}_1, \hat{T}), (\hat{S}_2, \hat{T})\}$ that overlap is at least the number of pairs of $\{(\hat{S}_1 \setminus \hat{S}_2, \hat{S}_2 \setminus \hat{S}_1), (\hat{S}_1 \setminus \hat{S}_2, \hat{T}), (\hat{S}_2 \setminus \hat{S}_1, \hat{T})\}$ that overlap.*

PROOF OF LEMMA 4.13. Let $\mathcal{F}$ be the initial family of F-witness bisets of edges in M. If no two bisets in $\mathcal{F}$ overlap, then $\mathcal{F}$ is the desired family. Otherwise, let $\hat{S}_{e_1}$ and $\hat{S}_{e_2}$ be two bisets that overlap. By Lemma 4.14 and Lemma 4.15, we can replace $\hat{S}_{e_1}$ and $\hat{S}_{e_2}$ with either $\hat{S}_{e_1} \cap \hat{S}_{e_2}$ and $\hat{S}_{e_1} \cup \hat{S}_{e_2}$ or by $\hat{S}_{e_1} \setminus \hat{S}_{e_2}$ and $\hat{S}_{e_2} \setminus \hat{S}_{e_1}$. The resulting family is an F-witness family of M that has fewer overlapping bisets. Thus, we can repeat this process until we get a non-overlapping F-witness family. □

Our approach is to use laminar families of witness bisets for the blue edges of F and the red edges of F' to count the regular and special vertices. Before we turn our attention to the counting arguments, we describe some properties of laminar families of witness bisets that we will need.

Laminar witness tree. For a *laminar* collection of bisets $\mathcal{L}$ defined on set V, let $\mathcal{L}^+$ denote the extended laminar family of $\mathcal{L}$, $\mathcal{L}^+ := \mathcal{L} \cup \{(V, V)\}$. We associate a tree $\mathcal{T}_{\mathcal{L}^+}$ with the family $\mathcal{L}^+$ as follows: The tree $\mathcal{T}_{\mathcal{L}^+}$ has a node $v_{\hat{S}}$ for each biset $\hat{S} \in \mathcal{L}^+$. For any two bisets $\hat{S}$ and $\hat{T}$ of $\mathcal{L}$ such that $\hat{S} \subset \hat{T}$, we add an edge from the node of $\mathcal{T}$ representing $\hat{S}$ to the node representing $\hat{T}$ iff there is no biset $\hat{X} \in \mathcal{L}$ such that $\hat{T} \subset \hat{X} \subset \hat{S}$. We view the tree $\mathcal{T}_{\mathcal{L}^+}$ as a rooted tree whose root is the node corresponding to the biset (V, V).

In the following, we consider a biuncrossbale function $h : \mathcal{P} \to \{0, 1\}$, a cover F for h, and a set of non-redundant edges $M \subseteq F$. We also fix a laminar family $\mathcal{L}$ of F-witness bisets for M and denote the tree associated with $\mathcal{L}^+$ by $\mathcal{T}$.

DEFINITION 4.16. A biset $\hat{S} \in \mathcal{L}^+$ **owns** $u \in V$ iff $\hat{S}$ is the minimal biset in $\mathcal{L}^+$ that contains u in its inner part.

PROPOSITION 4.17. *For each vertex $u \in V$, there is a unique biset in $\mathcal{L}^+$ that owns u.*

PROOF. Note that (V, V) contains u in its inner part, and thus there is a biset in $\mathcal{L} \cup \{(V, V)\}$ that owns u. Suppose for contradiction that two distinct bisets $\hat{X}$ and $\hat{Y}$ of $\mathcal{L} \cup \{(V, V)\}$ own u. Note that the set $X_{\text{in}} \cap Y_{\text{in}}$ is non-empty, since $u \in X_{\text{in}} \cap Y_{\text{in}}$. Since $\hat{X}$ and $\hat{Y}$ do not overlap, we must have $\hat{X} \subseteq \hat{Y}$ or $\hat{Y} \subseteq \hat{X}$. Therefore, one of $\hat{X}, \hat{Y}$ does not own u, which is a contradiction. □

PROPOSITION 4.18. *Let $\hat{C}$ be a minimal violated biset. Then all vertices of C_{in} are owned by the same biset in $\mathcal{L}^+$.*

PROOF. Let $\hat{C}$ be a minimal violated biset of h. If $\hat{C} \in \mathcal{L}^+$, then we are done. Consider any $\hat{Y} \in \mathcal{L}$; it is a violated biset. By Lemma 2.16, $\hat{C}$ and $\hat{Y}$ do not overlap. Thus, either $C_{\text{in}} \subseteq Y_{\text{in}}$ or $C_{\text{in}} \cap Y_{\text{in}}$ is empty. Consider the minimal biset $\hat{Y} \in \mathcal{L}^+$ such that $\hat{C} \subset \hat{Y}$ (since $\hat{C} \subset (V, V)$ such a biset exists). Then $\hat{Y}$ owns all vertices of C_{in}. □

It is convenient to abuse the notation and say that the node $v_{\hat{S}}$ of $\mathcal{T}$ owns u if $\hat{S}$ owns u. Additionally, we say that $\hat{S}$ owns $\hat{C}$ if it owns the inner part of $\hat{C}$. Consider an edge $e = uv$ in M, and assume that $\hat{S}, \hat{T} \in \mathcal{L}^+$ own u and v, respectively. The lemma below shows that either $v_{\hat{S}}$ is an ancestor of $v_{\hat{T}}$ or vice versa. It is possible for $\hat{T}$ to be a proper ancestor of $\hat{S}$ (that is, there is another biset $\hat{Y}$ in the family such that $\hat{S} \subset \hat{Y} \subset \hat{T}$) or vice versa.

LEMMA 4.19. *Let $e = uv$ be an edge of M. Let $\hat{S}_e$ be the F-witness biset of e in $\mathcal{L}$ and suppose $\hat{S}, \hat{T} \in \mathcal{L}^+$ own u and v, respectively.*

- *Then $\hat{S}_e = \hat{S}$ and $\hat{S} \subset \hat{T}$, or $\hat{S}_e = \hat{T}$ and $\hat{T} \subset \hat{S}$.*
- *Suppose, in addition, u is not contained in the boundary of any bisets in $\mathcal{L}^+$. If $\hat{T} \subset \hat{S}$, then $\hat{T}$ is a child of $\hat{S}$, that is, there is no $\hat{Y} \in \mathcal{L}^+$ such that $\hat{T} \subset \hat{Y} \subset \hat{S}$.*

PROOF. We consider the first part. Let w be the endpoint of e that is contained in the inner part of $\hat{S}_e$. Suppose for contradiction that $\hat{S}_e$ does not own w. Then there exists a biset $\hat{X} \subset \hat{S}_e$ in $\mathcal{L}$ such that $w \in X$. Note that $e \in \delta_F(\hat{X})$. However, $\hat{X}$ is an F-witness biset for an edge in $M \setminus \{e\}$, which is a contradiction. Therefore, $\hat{S}_e$ owns w. Thus, $\hat{S}_e = \hat{S}$ or $\hat{S}_e = \hat{T}$. Without loss of generality let us assume that $\hat{S}_e = \hat{S}$ and that $u \in S_{\text{in}}$. To complete the proof, we need to show that $\hat{S} \subset \hat{T}$. Note that we may assume that $\hat{T} \neq (V, V)$; otherwise, $\hat{S} \subset \hat{T}$ trivially holds. Moreover, $\hat{S} \neq \hat{T}$, since v is in $V \setminus S_{\text{out}}$ and $v \in T_{\text{in}}$. Since $\hat{S}$ is an F-witness biset for e, it follows that $\hat{T}$ is an F-witness biset for an edge of $M \setminus \{e\}$. Therefore, $e \notin \delta_F(\hat{T})$ and, since $v \in T_{\text{in}}$, we must have $u \in T_{\text{out}}$. Since $\hat{S}$ and $\hat{T}$ do not overlap and $u \in S \cap T_{\text{out}}$, either $\hat{S} \subset \hat{T}$ or $\hat{T} \subset \hat{S}$. However, since $v \in T_{\text{in}} \setminus S_{\text{out}}$, we cannot have $\hat{T} \subset \hat{S}$. Thus, $\hat{S} \subset \hat{T}$.

We now consider the second part where we assume that u is not in the boundary of any biset of $\mathcal{L}^+$. If $\hat{T} \subset \hat{S}$, then from the preceding part, we have $\hat{S}_e = \hat{T}$ and $v \in T_{\text{in}}$. Suppose for contradiction that there exists a biset $\hat{Y} \in \mathcal{L} \setminus \{\hat{S}, \hat{T}\}$ such that $\hat{T} \subset \hat{Y} \subset \hat{S}$. Since $\hat{Y}$ is an F-witness biset for an edge of $M \setminus \{e\}$, we must have $e \notin \delta_F(\hat{Y})$. Therefore, since $u \in S_{\text{in}} \setminus T_{\text{in}}$ and $v \in T_{\text{in}}$, $u \in Y_{\text{out}}$; since u is owned by $\hat{S}$ and not by $\hat{Y}$ it implies that $u \in \text{bd}(\hat{Y})$. However, by assumption u is not in the boundary of any biset of $\mathcal{L}^+$, which implies there is no such $\hat{Y}$. □

The following lemma is an important one that underlies the analysis:

LEMMA 4.20. *Let $e = uv$ be an edge of $F \setminus M$ such that $e \in \delta_F(\hat{C})$ for a minimal biset $\hat{C}$. Let u be the endpoint of e that is in $V \setminus C_{\text{out}}$. Let $\hat{S}$ be the biset of $\mathcal{L}^+$ that owns $\hat{C}$. If u is not contained in the boundary of any biset of $\mathcal{L}^+$, then $\hat{S}$ owns u.*

PROOF. First, we claim that $u \in S_{\text{in}}$. If $\hat{S} = (V, V)$, then the claim holds trivially. Therefore, we may assume that $\hat{S} \in \mathcal{L}$. Since $v \in C_{\text{in}}$ and $C_{\text{in}} \subset S_{\text{in}}$, we have $v \in S_{\text{in}}$. If $u \notin S_{\text{out}}$, then $e \in \delta(\hat{S})$ but $\hat{S}$ is an F-witness biset for some $e' \in M$ and $e \neq e'$, which is a contradiction.

Now suppose for contradiction that $\hat{S}$ does not own u. Then there is a biset $\hat{T} \in \mathcal{L}$ such that $\hat{T} \subset \hat{S}$, and $u \in T_{\text{in}}$. Since $\hat{T}$ is an F-witness biset for an edge in M and $u \in T_{\text{in}}$, we must have $v \in T_{\text{out}}$, which implies that $C_{\text{in}} \cap T_{\text{out}} \neq \emptyset$. By Lemma 2.16, $\hat{C}$ and $\hat{T}$ are non-overlapping, and, therefore, we must have $C_{\text{in}} \subseteq T_{\text{in}}$, which contradicts the fact that $\hat{S}$ owns $\hat{C}$. Hence, $\hat{S}$ owns u. □

PROPOSITION 4.21. *If $v_{\hat{S}}$ is a leaf of $\mathcal{T}$, then $\hat{S}$ owns a minimal violated biset.*

PROOF. Since $\hat{S}$ is a violated biset with respect to h, there exists a minimal violated biset $\hat{C} \in C$ such that $\hat{C} \subseteq \hat{S}$. Moreover, $\hat{S}$ is a minimal biset of $\mathcal{L}^+$; thus, $\hat{S}$ owns $\hat{C}$. □

Bijection between the edges of M and $\mathcal{T}$. We define the following bijection between the edges of $\mathcal{T}$ and the edges of M. Let e be an edge of M and let $\hat{S}_e$ be the witness biset for e. The node $v_{\hat{S}_e}$ has a parent $v_{\hat{T}}$ in $\mathcal{T}$, and we associate e with the edge $(v_{\hat{T}}, v_{\hat{S}_e})$ in $\mathcal{T}$. We say that the edge e *corresponds* to the edge $(v_{\hat{T}}, v_{\hat{S}_e})$.

PROPOSITION 4.22. *Let $\hat{C}$ be a minimal violated biset that is owned by $v_{\hat{S}} \in \mathcal{T}$. Then each edge $e \in \delta_M(\hat{C})$ whose endpoints are not in the boundary of any violated bisets, corresponds to an edge of $\mathcal{T}$ incident to $v_{\hat{S}}$.*

PROOF. Let $e = uv$ be an edge in $\delta_M(\hat{C})$ and let $u \in C_{\text{in}}$ and $v \notin C_{\text{out}}$. Since $\hat{S}$ owns $\hat{C}$, u is owned by $\hat{S}$. If $v \in V \setminus S_{\text{out}}$, then $\{e\} = \delta_M(\hat{S})$, which implies that $\hat{S} = \hat{S}_e$ and in this case e corresponds to $(v_{\hat{S}}, v_{\hat{T}})$ where $\hat{T}$ is the parent of $\hat{S}$. Otherwise, $v \in S_{\text{out}}$ and, since $e \in M$, it follows that $\{e\} = \delta_M(\hat{T})$ for some descendant of $\hat{S}$. Since u is not on the boundary of any violated biset by the assumption of the proposition, from the second part of Lemma 4.19, $\hat{T}$ is a child of $\hat{S}$, which implies that e corresponds to $(v_{\hat{T}}, v_{\hat{S}})$, which is incident to $v_{\hat{S}}$. □

Counting argument for regular vertices. Let $\mathcal{L}_b = \{\hat{S}_e \mid e$ is a blue edge in $F\}$ be a laminar family of F-witness bisets of the blue edges in F that is guaranteed by Lemma 4.13. Let $\mathcal{T}_b$ be the tree associated with $\mathcal{L}_b^+$.

Recall that each regular vertex is incident to at least one blue edge of F. Additionally, recall that F contains all the red edges of $K[Q']$. Therefore, for each regular vertex u, there is a red edge in F that is incident to u. Moreover, by Lemma 3.7, no violated biset contains a critical vertex in its boundary.

We charge each regular vertex u as follows: Let $\hat{C} \in C$ be a minimal violated biset for which there exists a red edge $wu \in \delta_G(\hat{C})$ such that $w \in C_{\text{in}}$ and $u \in V \setminus C_{\text{out}}$. Moreover, let $e = uv$ be a blue edge of F and let $\hat{S}_e \in \mathcal{L}_b$ be the F-witness biset for e. Suppose that $\hat{S}$ and $\hat{T}$ be the bisets that own u and v, respectively.

By Lemma 4.20 and the fact that u is not on the boundary of any violated biset, $\hat{S}$ owns $\hat{C}$. Additionally, by Lemma 4.19, we have either $\hat{S} \subset \hat{T}$ or $\hat{T} \subset \hat{S}$, and we consider each of these cases separately. Suppose that $\hat{S} \subset \hat{T}$. It follows from Lemma 4.19 that $\hat{S} = \hat{S}_e$. We charge u to $\hat{C}$. We refer to such a charge as a *parent charge*. Next, suppose that $\hat{T} \subset \hat{S}$. By Lemma 4.19 and the fact that u is not on the boundary of any violated biset, $v_{\hat{T}}$ is a child of $v_{\hat{S}}$. Since, by Proposition 4.21, each leaf of $\mathcal{T}_b$ owns a biset of C, there is a descendant of $v_{\hat{T}}$ (possibly $v_{\hat{T}}$ itself) that owns a biset of C. Let $v_{\hat{X}}$ be the closest such descendant, i.e., a descendant whose distance to $v_{\hat{T}}$ is minimized. (If there are several descendants whose distance to $v_{\hat{T}}$ is minimum, then we pick one of them arbitrarily.) We charge u to one of the bisets of C that $v_{\hat{X}}$ owns. We refer to this charge as a *subtree charge*, since u is charged in a subtree rooted at a child of the node $v_{\hat{S}}$ that owns u.

LEMMA 4.23. *There is at most one parent charge to each biset $\hat{C} \in C$.*

PROOF. Let $\hat{S}$ be the biset of $\mathcal{L}_b^+$ that owns $\hat{C}$. Suppose that $\hat{C}$ incurs a parent charge from a vertex u. Then u is in S and there is a blue edge $e = uv \in F$ such that $\hat{S} = \hat{S}_e$. Since $\hat{S}$ is an F-witness biset for exactly one edge, there is at most one parent charge to $\hat{C}$.   □

LEMMA 4.24. *There is at most one subtree charge to each biset $\hat{C} \in C$.*

PROOF. Let $\hat{X}$ be the biset of $\mathcal{L}_b^+$ that owns $\hat{C}$. Suppose that there is a descendant charge to $\hat{C}$ corresponding to a vertex u_1. Then there is a blue edge $e_1 = u_1 v_1$ whose endpoints are owned by $\hat{S}_1$ and $\hat{T}_1$ (respectively) and $v_{\hat{T}_1}$ is a child of $v_{\hat{S}_1}$. By the way, we choose $\hat{X}$, $v_{\hat{X}}$ and $v_{\hat{S}_1}$ are the only nodes on the path in $\mathcal{T}_b$ from $v_{\hat{S}_1}$ to $v_{\hat{X}}$ that own a biset of C. Moreover, $\hat{T}_1$ is the witness biset of e_1.

Suppose for contradiction that there is another descendant charge to $\hat{C}$ from $u_2 \neq u_1$. Let $\hat{S}_2$ and $\hat{T}_2$ be the bisets of $\mathcal{L}^+$ that own u_2 and v_2 (respectively). By the same argument as above, $v_{\hat{S}_2}$ is a child of $v_{\hat{T}_2}$ and there is no node in the path from $v_{\hat{S}_2}$ to $v_{\hat{X}}$ other than its endpoints that owns a biset of C. Moreover, $\hat{T}_2$ is the witness biset of $e_2 = u_2 v_2$ (which is different from e_1).

Since $\hat{T}_1$ and $\hat{T}_2$ are distinct bisets of $\mathcal{L}_b^+$ that both contain $\hat{X}$, either $\hat{T}_1 \subset \hat{T}_2$ or $\hat{T}_2 \subset \hat{T}_1$. Moreover, since for each $i \in \{1, 2\}$, $\hat{X} \subset \hat{S}_i$, $\hat{S}_i \not\subset \hat{T}_{2-i}$ and $v_{\hat{T}_i}$ is a child of $v_{\hat{S}_i}$, we must have $\hat{S}_1 = \hat{S}_2$. Thus, $v_{\hat{T}_1}$ and $v_{\hat{T}_2}$ are children of $\hat{S}$, which contradicts the fact that either $\hat{T}_1 \subset \hat{T}_2$ or $\hat{T}_2 \subset \hat{T}_1$. Hence, there is at most one subtree charge to each minimal violated biset $\hat{C}$.   □

PROOF OF LEMMA 4.9. By Lemma 4.23 and 4.24, each biset of C is charged at most twice and thus the number of regular vertices is at most $2|C|$.   □

Counting argument for special vertices. Recall that F' is an edge-minimal cover of h'. Moreover, a critical vertex v is special only if there is a red edge e incident to v such that $e \in \delta_{F'}(\hat{C})$ for a minimal violated biset $\hat{C}$ where $v \in V \setminus C'$. Thus, the total number of special vertices is upper bounded by $\sum_{\hat{C} \in C} |\delta_{F'}(\hat{C})|$. Next, we adopt the argument of Jain et al. [25] to show that, for any edge-minimal cover F', $\sum_{\hat{C} \in C} |\delta_{F'}(\hat{C})|$ is at most $2|C|$. Let $\mathcal{L}_r$ be the laminar family of witness bisets of the red edges with respect to F' and let $\mathcal{T}_r$ be the tree representation of $\mathcal{L}_r^+$. Let $\mathcal{A}$ denote the set of vertices in $\mathcal{T}_r$ that own a minimal violated biset. By Proposition 4.22, $\sum_{\hat{C} \in C} |\delta_{F'}(\hat{C})| \leq \sum_{v \in \mathcal{A}} \deg(v)$. Here, $\deg(v)$ refers to the degree of node v in $\mathcal{T}_r$. Note that by Proposition 4.21, all leaf vertices are in $\mathcal{A}$ and there is at most one node in $V(\mathcal{T}_r) \setminus \mathcal{A}$ with degree less than 2; the root node. Hence,

$$\sum_{\hat{C} \in C} |\delta_{F'}(\hat{C})| \leq \sum_{v \in \mathcal{A}} \deg(v)$$

$$\leq \sum_{v \in V(\mathcal{T}_r)} \deg(v) - \sum_{v \in V(\mathcal{T}_r) \setminus \mathcal{A}} \deg(v)$$

$$\leq 2(|V(\mathcal{T}_r)| - 1) - 2(|V(\mathcal{T}_r)| - |\mathcal{A}| - 1)$$

$$\leq 2|\mathcal{A}| \leq 2|C|.$$

Thus, we can upper bound the number of special vertices by $2|C|$, which proves Lemma 4.10. We remark that some of the regular vertices are counted in this step as well, but this can only help us.

5 ALGORITHM FOR {0, 1, 2}-VC-SNDP

In this section, we prove the following theorem family of graphs.

THEOREM 5.1. *There is an $O(1)$-approximation for node-weighted VC-SNDP when the requirements are in $\{0, 1, 2\}$ and the input graph belongs to a proper minor-closed family of graphs.*

We construct a solution in two stages. In the first stage, we use an algorithm for node-weighted Steiner forest to find a set $X_1 \subseteq V$ of nodes such that $G[X_1]$ connects each pair uv with $r(u, v) \geq 1$. A constant factor approximation for this in proper minor-closed families of graphs follows from prior work that we already discussed [13, 30]. Letting OPT denote the weight of an optimum solution for the initial instance, it is straightforward to see that $w(X_1) = O(\text{OPT})$. Let F be the edge set of the graph $E[X_1]$. In the second stage, we augment X_1 to 2-connect pairs (s, t) with connectivity requirement 2. For the second stage, as with Elem-SNDP, we define a $\{0, 1\}$-biuncrossable function h and a graph $G' = (V, E \setminus F)$. Let $\mathcal{P}_{vc}$ be the collection of all bisets over V. The requirement function $r_v : \mathcal{P}_{vc} \to \{0, 1, 2\}$ for each biset $\hat{S}$ is defined as the maximum connectivity requirement over all pair of vertices that are separated by $\hat{S}$. Let $h : \mathcal{P}_{vc} \to \{0, 1\}$ be the function such that $h(\hat{S}) = 1$ iff $r_v(\hat{S}) = 2$ and $|\delta_F(\hat{S})| + |\text{bd}(\hat{S})| = 1$. By Menger's theorem on vertex connectivity, a feasible cover of h_v together with F is a feasible solution for the VC-SNDP instance. For the second stage, we are only interested in those pairs (s, t) such that $r(s, t) = 2$ while s and t are only 1-connected in $G[X_1]$. We call a vertex u a terminal for the second stage if it participates in such a pair.

PROPOSITION 5.2 (LEMMA 5.1 IN [15]). *The function r_v is biuncrossable.*

PROPOSITION 5.3. *The function r_v is bimaximal.*

PROOF. Let $\hat{S}, \hat{T} \in \mathcal{P}_{vc}$ and let (s, t) be a pair of terminals that have the maximum connectivity requirement among all terminal pairs separated by $\hat{S} \cup \hat{T}$, i.e., $r_v(\hat{S} \cup \hat{T}) = r(s, t)$. Since $s \in S_{in} \cup T_{in}$, we have $s \in S_{in}$ or $s \in T_{in}$; without loss of generality, assume $s \in S_{in}$. Since $t \in V \setminus (S_{out} \cup T_{out})$, the pair (s, t) is separated by $\hat{S}$ and thus $r_v(\hat{S} \cup \hat{T}) \leq r_v(\hat{S}) \leq \max\{r_v(\hat{S}), r_v(\hat{T})\}$.   □

LEMMA 5.4. *The function h is biuncrossable.*

PROOF. By Lemma 2.12, $|\delta_F(.)|$ is bisubmodular and by Proposition 2.9, $|\text{bd}(.)|$ is bisubmodular. If $h(\hat{S}) = h(\hat{T}) = 1$, then $r_v(\hat{S}) = r_v(\hat{T}) = 2$, and $|\delta_F(\hat{S})| + |\text{bd}(\hat{S})| = |\delta_F(\hat{T})| + |\text{bd}(\hat{T})| = 1$.

Since r_v is biuncrossable (Proposition 5.2),

$$r_v(\hat{S}) + r_v(\hat{T}) \leq \max\{r_v(\hat{S} \cap \hat{T}) + r_v(\hat{S} \cup \hat{T}), r_v(\hat{S} \setminus \hat{T}) + r_v(\hat{T} \setminus \hat{S})\}.$$

Without loss of generality, assume that $r_v(\hat{S}) + r_v(\hat{T}) \leq r_v(\hat{S} \cap \hat{T}) + r_v(\hat{S} \cup \hat{T})$, which by the upper bound of 2 on the connectivity requirements implies that $r_v(\hat{S} \cap \hat{T}) = r_v(\hat{S} \cup \hat{T}) = 2$.

Moreover, since $|\delta_F(.)|$ and $|\text{bd}(.)|$ are both bisubmodular, $|\delta_F(\hat{S})| + |\text{bd}(\hat{S})| + |\delta_F(\hat{T})| + |\text{bd}(\hat{T})| \geq |\delta_F(\hat{S} \cap \hat{T})| + |\text{bd}(\hat{S} \cap \hat{T})| + |\delta_F(\hat{S} \cup \hat{T})| + |\text{bd}(\hat{S} \cup \hat{T})|$. Since the edge set F connects all pair of terminals with non-zero connectivity requirements, both $|\delta_F(\hat{S} \cap \hat{T})| + |\text{bd}(\hat{S} \cap \hat{T})|, |\delta_F(\hat{S} \cup \hat{T})| + |\text{bd}(\hat{S} \cup \hat{T})|$ are at least 1. Hence, $|\delta_F(\hat{S} \cap \hat{T})| + |\text{bd}(\hat{S} \cap \hat{T})| = |\delta_F(\hat{S} \cup \hat{T})| + |\text{bd}(\hat{S} \cup \hat{T})| = 1$.

The other case holds similarly and thus h is biuncrossable.   □

Next, we prove analogues of Lemmas 3.7 and 3.8, which show that h satisfies the key properties that allowed us to use and analyze the primal-dual algorithm from Section 4.

LEMMA 5.5. *For any $X \supset X_1$ let $H'_X = (V, E_{G'}[X])$ be a subgraph of G'. Suppose $\hat{C}$ is a violated biset of H'_X with respect to h. Then $\text{bd}(\hat{C}) \subseteq X$.*

PROOF. We use H' in place of H'_X for ease of notation. Suppose $\hat{C}$ is a violated biset with respect to h in H' and there is a vertex $u \in \text{bd}(\hat{C})$ such that $u \notin X$. Consider the biset $\hat{C}_1 = (C, \text{bd}(\hat{C}) \setminus \{u\})$. Note that u is not a terminal and hence $r_v(\hat{C}_1) = r_v(\hat{C}) = 2$. Since $\hat{C}$ is a violated biset in H', we have $|\text{bd}(\hat{C})| + \delta_{H'}(\hat{C}) = 1$ and, since $|\text{bd}(\hat{C})| \geq 1$ (since u is in the boundary), we have $\delta_{H'}(\hat{C}) = \emptyset$.

Consider the bisets $\hat{C}$ and $\hat{C}_1$ in the graph $H_1 = (V, E[X_1])$. Since $r_v(\hat{C}) = 2$ and $\hat{C}$ is violated in H', we have $|\text{bd}(\hat{C})| + |\delta_{H_1}(\hat{C})| = 1$ but then $|\text{bd}(\hat{C}_1)| + |\delta_{H_1}(\hat{C}_1)| = 0$, since u has no edges incident to it in H_1. Since $r_v(\hat{C}_1) = 2$ this implies that H_1 is not a feasible solution to 1-connect the terminals in the first stage, a contradiction.   □

LEMMA 5.6. *For any $X \supset X_1$ let $H'_X = (V, E_{G'}[X])$ be a subgraph of G'. Suppose $\hat{C}$ is a minimal violated biset of H'_X with respect to h. Then the following properties hold:*

- $C_{\text{out}} \subseteq X$.
- $G[C_{\text{in}}]$ *is a connected subgraph of G.*

PROOF. For ease of notation, we let H' denote the graph H'_X. Since $\hat{C}$ is a violated biset in H', we have $h(\hat{C}) = 1$ and $|\delta_{H'}(\hat{C})| = 0$. From the definition of h, $r_v(\hat{C}) = 2$ and $|\text{bd}(\hat{C})| + |\delta_{G[X]}(\hat{C})| = 1$. Suppose there is a vertex $u \in C_{\text{out}}$ such that $u \notin X$. By Lemma 5.5, $u \in C_{\text{in}}$. First, u is not a terminal, since all terminals are in X_1 (and hence in X). Second u is an isolated vertex in H', since the only edges in H' are between nodes in X. Consider the biset $\hat{C}_1 = (C_{\text{in}} \setminus \{u\}, C_{\text{out}} \setminus \{u\})$ obtained from $\hat{C}$ by removing u from both inner and outer parts of $\hat{C}$. Since u is not a terminal, we have $r_v(\hat{C}) = r_v(\hat{C}_1)$. And, since u is isolated in H', we have $\delta_{H'}(\hat{C}_1) = \delta_{H'}(\hat{C})$, and $\text{bd}(\hat{C}_1) \subseteq \text{bd}(\hat{C})$. These facts imply that $\hat{C}_1) = 1$ and $\hat{C}_1$ is a violated biset in H'. This contradicts minimality of $\hat{C}$. Therefore, $C_{\text{out}} \subseteq X$.

Next, we prove that $G[C_{\text{in}}]$ is connected. For sake of contradiction, suppose it is not. Let $\overline{C}_1, \overline{C}_2$ be two non-empty sets that partition C_{in} such that there is no edge between $\overline{C}_1$ and $\overline{C}_2$ in G; such a partition exists if $G[C_{\text{in}}]$ is not connected. Note that $E_{H'}(\overline{C}_1, \overline{C}_2) = \emptyset$, since H' is a subgraph of G. Define $\hat{C}_1 = (\overline{C}_1, \overline{C}_1 \cup \text{bd}(\hat{C}))$ and $\hat{C}_2 = (\overline{C}_2, \overline{C}_2 \cup \text{bd}(\hat{C}))$. Since r_v is bimaximal, $r_v(\hat{C}) \leq \max\{r_v(\hat{C}_1), r_v(\hat{C}_2)\}$. Thus, without loss of generality, we can assume that $r_v(\hat{C}_1) \geq r_v(\hat{C}) \geq 2$. Since $E_{H'}(\overline{C}_1, \overline{C}_2) = \emptyset$, we have $\delta_{H'}(\hat{C}_1) \subseteq \delta_{H'}(\hat{C})$. Since $\text{bd}(\hat{C}_1) = \text{bd}(\hat{C})$ and $\hat{C}$ was a violated biset it follows that $\hat{C}_1$ is also a violated biset with respect to h in H'. This contradicts the minimality of $\hat{C}$.   □

For any $X \supseteq X_1$ finding minimal violated bisets of h with respect to the graph H'_X can be easily done in polynomial time via Max-Flow algorithms.

The function h satisfies the same properties as those that arise in the augmentation framework for Elem-SNDP and, hence, we can apply the primal-dual algorithm and analysis as captured by Theorem 4.1. The algorithm outputs a node set Q such that $G[Q]$ covers h and $w(Q \setminus X) = O(\text{OPT})$. Since $w(X) = O(\text{OPT})$, we have that $w(Q \cup X) = O(\text{OPT})$. For planar graphs, we can obtain a concrete upper bound of 13OPT using the 3-approximation for the first stage and a 10-approximation for the second stage.

6 CONCLUDING REMARKS

We obtained approximation algorithms for node-weighted network design in planar and minor-closed families of graphs when the connectivity requirements are larger than one. We built upon the insights from [13] as well as prior work via the augmentation framework for SNDP. The analysis of the primal-dual algorithm that we present is probably not tight and it would be interesting to obtain the tightest bound one can prove for the algorithm. It may be possible to borrow ideas from [4] and alter the algorithm to obtain improved approximation ratios. For general VC-SNDP, we obtain an improvement over the general graph case via our algorithm for EC-SNDP and a black-box reduction of [11]. For two important special cases of VC-SNDP, namely, Rooted-VC-SNDP and Subset-VC-SNDP, $O(k \log k)$-approximations are known in the edge-weighted case [29, 32] and the node-weighted case requires an additional $O(\log n)$-factor. It would be interesting

to show that this additional factor is unnecessary in planar graphs—we note that the results in [29, 32] are based on the augmentation framework and hence some of our ideas may be applicable.

Finally, it is an interesting question whether there is an $O(1)$-approximation for node-weighted EC-SNDP and other network design problems in planar graphs. Is the dependence on k necessary? Recall that, for general graphs, we expect a dependence on k via the hardness reduction from the k-Densest-Subgraph problem [31]. However, k-Densest-Subgraph is constant-factor approximable in planar graphs.

REFERENCES

[1]   A. Agrawal, P. Klein, and R. Ravi. 1995. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.* 24, 3 (1995), 440–456.

[2]   M. Bateni, M. Hajiaghayi, and V. Liaghat. 2013. Improved approximation algorithms for (budgeted) node-weighted Steiner problems. In *International Colloquium on Automata, Languages, and Programming*. Springer, 81–92.

[3]   M. Bateni, M. Hajiaghayi, and D. Marx. 2011. Approximation schemes for Steiner forest on planar graphs and graphs of bounded treewidth. *J. ACM* 58, 5 (2011), 21.

[4]   P. Berman and G. Yaroslavtsev. 2012. Primal-dual approximation algorithms for node-weighted network design in planar graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 50–60.

[5]   G. Borradaile, P. Klein, and C. Mathieu. 2009. An O(N Log N) approximation scheme for Steiner tree in planar graphs. *ACM Trans. Algor.* 5, 3 (July 2009). DOI : https://doi.org/10.1145/1541885.1541892

[6]   J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. 2010. An improved LP-based approximation for Steiner tree. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*. ACM, 583–592.

[7]   J. Byrka, F. Grandoni, T. Rothvoss, and L. Sanità. 2013. Steiner tree approximation via iterative randomized rounding. *J. ACM* 60, 1 (2013), 6.

[8]   C. Chekuri, A. Ene, and A. Vakilian. 2012. Node-weighted network design in planar and minor-closed families of graphs. In *Automata, Languages, and Programming*. Springer, 206–217.

[9]   C. Chekuri, A. Ene, and A. Vakilian. 2012. Prize-collecting survivable network design in node-weighted graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 98–109.

[10] J. Cheriyan, S. Vempala, and A. Vetta. 2006. Network design via iterative rounding of setpair relaxations. *Combinatorica* 26, 3 (2006), 255–275.

[11] J. Chuzhoy and S. Khanna. 2012. An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design. *Theor. Comput.* 8 (2012), 401–413.

[12] Vincent Cohen-Addad, Éric Colin de Verdière, Philip N. Klein, Claire Mathieu, and David Meierfrankenfeld. 2016. Approximating connectivity domination in weighted bounded-genus graphs. In *Proceedings of the 48th ACM Symposium on Theory of Computing*. 584–597.

[13] E. D. Demaine, M. Hajiaghayi, and P. N. Klein. 2014. Node-weighted Steiner tree and group Steiner tree in planar graphs. *ACM Trans. Algor.* 10, 3 (2014), 13.

[14] A. Ene, N. Korula, and A. Vakilian. 2013. Connected domatic packings in node-capacitated graphs. *CoRR* abs/1305.4308 (2013).

[15] L. Fleischer, K. Jain, and D. P. Williamson. 2006. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *J. Comput. Syst. Sci.* 72, 5 (2006), 838–867.

[16] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. 2020. Approximation schemes via width/weight trade-offs on minor-free graphs. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2299–2318.

[17] T. Fukunaga. 2017. Spider covers for prize-collecting network activation problem. *ACM Trans. Algor.* 13, 4 (2017), 49.

[18] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. B. Shmoys, E. Tardos, and D. P. Williamson. 1994. Improved approximation algorithms for network design problems. In *Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithms*. 223–232.

[19] M. X. Goemans and D. P. Williamson. 1995. A general approximation technique for constrained forest problems. *SIAM J. Comput.* 24, 2 (1995), 296–317.

[20] M. X. Goemans and D. P. Williamson. 1997. The primal-dual method for approximation algorithms and its application to network design problems. In *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., 144–191.

[21] M. X. Goemans and D. P. Williamson. 1998. Primal-dual approximation algorithms for feedback problems in planar graphs. *Combinatorica* 18, 1 (1998), 37–59.

[22] S. Guha and S. Khuller. 1999. Improved methods for approximating node weighted Steiner trees and connected dominating sets. *Inf. Comput.* 150, 1 (1999), 57–74.

[23] A. Gupta and J. Könemann. 2011. Approximation algorithms for network design: A survey. *Surv. Oper. Res. Manag. Sci.* 16, 1 (2011), 3–20.

[24] K. Jain. 2001. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica* 21, 1 (2001), 39–60.

[25] K. Jain, I. I. Mandoiu, V. V. Vazirani, and D. P. Williamson. 2002. A primal-dual schema based approximation algorithm for the element connectivity problem. *J. Algor.* 45, 1 (2002), 1–15.

[26] P. N. Klein and R. Ravi. 1995. A nearly best-possible approximation algorithm for node-weighted Steiner trees. *J. Algor.* 19, 1 (1995), 104–115.

[27] G. Kortsarz and Z. Nutov. 2010. Approximating minimum cost connectivity problems. In *Proceedings of the Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[28] A. V. Kostochka. 1984. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica* 4, 4 (1984), 307–316.

[29] B. Laekhanukit. 2015. An improved approximation algorithm for the minimum cost subset k-connected subgraph problem. *Algorithmica* 72, 3 (2015), 714–733.

[30] C. Moldenhauer. 2013. Primal-dual approximation algorithms for node-weighted Steiner forest on planar graphs. *Inf. Comput.* 222 (2013), 293–306.

[31] Z. Nutov. 2010. Approximating Steiner networks with node-weights. *SIAM J. Comput.* 39, 7 (2010), 3001–3022.

[32] Z. Nutov. 2012. Approximating minimum-cost connectivity problems via uncrossable bifamilies. *ACM Trans. Algor.* 9, 1 (2012), 1.

[33] Z. Nutov. 2012. Approximating Steiner network activation problems. In *Proc. of LATIN*.

[34] Z. Nutov. 2018. Erratum: Approximating minimum-cost connectivity problems via uncrossable bifamilies. *ACM Trans. Algor.* 14, 3 (June 2018). DOI : https://doi.org/10.1145/3186991

[35] D. Panigrahi. 2011. Survivable network design problems in wireless networks. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1014–1027.

[36] A. Vakilian. 2013. *Node-weighted Prize-collecting Survivable Network Design Problems*. Master's thesis. University of Illinois, Urbana-Champaign, IL.

[37] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. 1995. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica* 15, 3 (1995), 435–454.