SIAM J. COMPUT.
Vol. 39, No. 7, pp. 3001–3022

APPROXIMATING STEINER NETWORKS WITH NODE-WEIGHTS*

ZEEV NUTOV†

Abstract. The (undirected) Steiner Network problem is as follows: given a graph $G = (V, E)$ with edge/node-weights and edge-connectivity requirements $\{r(u, v) : u, v \in U \subseteq V\}$, find a minimum-weight subgraph H of G containing U so that the uv-edge-connectivity in H is at least $r(u, v)$ for all $u, v \in U$. The seminal paper of Jain [*Combinatorica*, 21 (2001), pp. 39–60], and numerous papers preceding it, considered the Edge-Weighted Steiner Network problem, with weights on the edges only, and developed novel tools for approximating minimum-weight edge-covers of several types of set functions and families. However, for the Node-Weighted Steiner Network (NWSN) problem, nontrivial approximation algorithms were known only for 0, 1 requirements. We make an attempt to change this situation by giving the first nontrivial approximation algorithm for NWSN with arbitrary requirements. Our approximation ratio for NWSN is $r_{\max} \cdot O(\ln |U|)$, where $r_{\max} = \max_{u,v \in U} r(u, v)$. This generalizes the result of Klein and Ravi [*J. Algorithms*, 19 (1995), pp. 104–115] for the case $r_{\max} = 1$. We also give an $O(\ln |U|)$-approximation algorithm for the node-connectivity variant of NWSN (when the paths are required to be internally disjoint) for the case $r_{\max} = 2$. Our results are based on a much more general approximation algorithm for the problem of finding a minimum node-weighted edge-cover of an *uncrossable set-family*. Finally, we give evidence that a polylogarithmic approximation ratio for NWSN with large $r_{\max}$ might not exist even for $|U| = 2$ and unit weights.

Key words. Steiner networks, node-weights, intersecting families, approximation algorithms

AMS subject classifications. 68W25, 05C40, 05C85

DOI. 10.1137/080729645

1. Introduction.

1.1. Motivation, problem definition, and previous work.

Network design problems require finding a minimum-weight (sub)network that satisfies prescribed properties, often connectivity requirements. Classic examples with 0, 1 connectivity requirements are Shortest Path, Minimum Spanning Tree, Minimum Steiner Tree/Forest, and others. Examples of problems with high connectivity requirements are Min-Cost k-Flow, k-Edge/Node-Connected Spanning Subgraph, Steiner Network, and others.

Two main types of weights are considered in the literature: the edge-weights and the node-weights. We consider the latter, which are usually more general than the former. For most undirected network design problems, a simple reduction transforms edge-weights to node-weights, but the inverse is usually not true. The study of network design problems with node-weights is well motivated and established from both theoretical as well as practical considerations; cf. [18, 14, 23, 4, 21]. For example, in telecommunication networks, expensive equipment, such as routers, switches, and transmitters, is located at the nodes of the network, and thus it is natural to model these problems by assigning weights to the nodes and/or to the edges, rather than to the edges only.

In directed graphs, it is often possible to reduce the node-weights case to the edge-weights case via an approximation ratio preserving reduction. However, this is usually not so for undirected graphs, and an attempt to transform an undirected problem

*Received by the editors July 9, 2008; accepted for publication (in revised form) March 23, 2010; published electronically June 9, 2010. A preliminary version of this paper appeared as [25].
http://www.siam.org/journals/sicomp/39-7/72964.html
†Department of Computer Science, The Open University of Israel, Raanana 43107, Israel (nutov@openu.ac.il).

into a directed one typically results in a problem which is significantly harder to approximate. For example, on undirected graphs, for Steiner Forest a 2-approximation is known for edge-weights [1], and an $O(\log n)$-approximation is known for node-weights, and this ratio is tight [18], while the *directed* variant does not admit a polylogarithmic ratio unless NP⊆Quasi(P) [6]. In fact, the best known ratio for the directed variant is much worse than this lower bound; see [9].

Let $\lambda_H(u,v)$ denote the maximum number of edge-disjoint uv-paths in a graph H. We consider the following fundamental problem on undirected graphs.

Node-Weighted Steiner Network (NWSN).
Instance: A graph $G=(V,E)$ with node-weights $\{w(v) : v \in V\}$ and edge-connectivity requirements $\{r(u,v) : u,v \in U \subseteq V\}$.
Objective: Find a minimum-weight subgraph H of G containing U so that

$$\lambda_H(u,v) \geq r(u,v) \quad \text{for all } u,v \in U. \tag{1}$$

Let $r_{\max} = \max_{u,v\in U} r(u,v)$ be the maximum requirement. The Edge-Weighted Steiner Network problem was studied extensively, starting from the first 2-approximation algorithm of Agrawal, Klein, and Ravi [1] for $r_{\max} = 1$ (see Goemans and Williamson [12] for a more general algorithm and simpler proof), continuing with the $2r_{\max}$-approximation of Williamson et al. [30] and the $O(\ln r_{\max})$-approximation of Goemans et al. [11], and ending with the seminal 2-approximation of Jain [16]. See surveys in [13, 17, 20] on approximation algorithms for various connectivity problems.

However, for the node-weighted version, NWSN, nontrivial approximation algorithms were known only for $r_{\max} = 1$. The first approximation algorithm for NWSN with $r_{\max} = 1$ due to Klein and Ravi [18] appeared in 1995, at the same time as the 2-approximation of Agrawal, Klein, and Ravi [1] for the edge-weighted case with $r_{\max} = 1$. The Klein–Ravi [18] algorithm uses a greedy approach. Using "spider-decomposition" of trees, they proved that iteratively adding spiders (subtrees with at most one node of degree ≥ 3) that minimize a certain ratio (the weight of the spider over the number of "minimal deficient sets" it connects minus 1) is a $2H(|U|)$-approximation algorithm, where $H(n) = \sum_{i=1}^{n} 1/i = O(\ln n)$ is the nth harmonic number. The approximation ratio was improved by Guha and Khuller [14] to $(1.35+\varepsilon)H(|U|)$ using a slight generalization of spiders. These ratios are nearly tight, as the case $r_{\max} = 1$ of NWSN generalizes the Set-Cover problem and thus has a $(1-\varepsilon)\ln|U|$-approximation threshold [7]. However, unlike the case of edge-weights, for node-weights almost no progress has been made since the Klein–Ravi paper [18]: no approximation algorithm was known for NWSN with $r_{\max} > 1$, not even for the case $r_{\max} = 2$.

1.2. Our results. We give the first nontrivial algorithm for NWSN with arbitrary requirements.

THEOREM 1.1. NWSN *admits a* $6r_{\max} \cdot H(|U|)$*-approximation algorithm.*

The approximation ratio in Theorem 1.1 is tight (up to a constant factor) if $r_{\max}$ is "small" (usually, $r_{\max} \leq 3$ in practical networks), but may seem weak if $r_{\max}$ is large. We give the first evidence that a polylogarithmic approximation algorithm for NWSN may not exist even for very simple instances. Let the Node-Weighted k-Flow (NWkF) problem be the restriction of NWSN to instances with $U = \{s,t\}$ and $r(s,t) = k$. It is not hard to see that a ρ-approximation for NWkF implies a ρ-approximation for the Set-Cover problem, but this implies only an $\Omega(\ln n)$-approximation threshold. To obtain evidence that NWkF, and thus also NWSN, may not admit a polylogarithmic approximation ratio, we give a reduction from the following extensively studied problem to unit weight NWkF. For an edge set E on V and $X \subseteq V$, let $E(X)$ denote the set of edges in E with both end-nodes in X.

Densest ℓ-Subgraph.
Instance: A graph $G = (V, E)$ and an integer ℓ.
Objective: Find $X \subseteq V$ with $|X| \leq \ell$ and $|E(X)|$ maximum.

No polylogarithmic approximation ratio is known for the Densest ℓ-Subgraph problem, although it has been studied extensively. The currently best known ratio for the problem due to Bhaskara et al. [2] is $O(|V|^{-1/4-\varepsilon})$ in time $O(|V|^{1/\varepsilon})$ (see also [8] for an earlier $O(|V|^{-1/3})$ ratio). This is so even for the case of bipartite graphs, which up to a constant factor is as hard to approximate as the general case. We prove the following theorem.

THEOREM 1.2. *Suppose that* NWkF *admits a ρ-approximation algorithm. Then the following hold:*

- *The* Set-Cover *problem admits a ρ-approximation algorithm.*
- Densest ℓ-Subgraph *on bipartite graphs admits a $1/(2\rho^2)$-approximation algorithm.*

Remark. It is shown in [15] that *directed* NWkF cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any fixed $\varepsilon > 0$ unless NP $\subseteq$ Quasi(P); for edge-weights, this case is in P. On the other hand, the "augmentation" version of NWkF that seeks to find a minimum node-weight augmenting edge-set to increase the st-edge-connectivity by 1 is reducible to the Shortest Path problem and thus is solvable in polynomial time; see section 6. This implies a k-approximation algorithm for NWkF. Also, NWkF with node-disjoint paths is easily reducible to the Min-Cost k-Flow problem and thus is solvable in polynomial time.

We also consider the node-connectivity version of NWSN, when the paths are required to be internally node-disjoint. The edge-weighted version with internally disjoint paths is usually referred to as the Survivable Network Design (SND) problem. SND does not admit a polylogarithmic approximation algorithm unless NP$\subseteq$Quasi(P) [19], and this is so even if the input graph G is complete with edge-weights in $\{0, 1\}$ [26, 22]. However, Ravi and Williamson [29] showed that the $\{0, 1, 2\}$-SND, when $r_{\max} \leq 2$, admits a 3-approximation algorithm using the primal-dual method; the ratio was improved to 2 by Fleischer, Jain, and Williamson [10] using the iterative rounding method. We consider the node-weighted version NWSND of SND, and specifically the $\{0, 1, 2\}$-NWSND, and prove the following theorem.

THEOREM 1.3. $\{0, 1, 2\}$-NWSND *admits an $O(\ln n)$-approximation algorithm.*

Theorems 1.1 and 1.3 are just applications of a more general approximation algorithm for finding a minimum "node-weighted" (edge-)cover of an extensively studied type of set-family. We need some definitions to present this result. For a graph $H = (V, I)$ and $X \subseteq V$ let $\deg_H(X) = \deg_I(X)$ denote the *degree of* X, namely, the number of edges in I with exactly one end-node in X. For an edge-set I on V, let $V(I) = \bigcup_{uv \in I}\{u, v\}$ denote the set of end-nodes of the edges in I. Given node-weights $\{w(v) : v \in V\}$, let $w(I) = w(V(I))$ be the *node-weight* of I.

DEFINITION 1.1. *Let $\mathcal{F} \subseteq 2^V$ be a set-family of subsets of a ground-set V.*

- *$\mathcal{F}$ is* uncrossable *if $X \cap Y, X \cup Y \in \mathcal{F}$ or $X - Y, Y - X \in \mathcal{F}$ for any $X, Y \in \mathcal{F}$.*
- *An edge set I on V* covers *$\mathcal{F}$ (or I is an $\mathcal{F}$-*cover*) if $\deg_I(X) \geq 1$ for every $X \in \mathcal{F}$.*

We consider the following general problem.

Node-Weighted Set-Family (Edge-)Cover (NWSFC).
Instance: A set-family $\mathcal{F}$, an edge-set E on V, and node-weights $\{w(v) : v \in V\}$.
Objective: Find a minimum node-weight $\mathcal{F}$-cover $I \subseteq E$.

We give an $O(\ln |V|)$-approximation algorithm for the problem of finding a minimum node-weight cover of an uncrossable family $\mathcal{F}$, but its polynomial implementation requires that certain queries related to $\mathcal{F}$ can be answered in polynomial time. Given an edge-set I on V (I is a partial cover of $\mathcal{F}$), the *residual family $\mathcal{F}_I$ of $\mathcal{F}$* (w.r.t. I) consists of all members of $\mathcal{F}$ that are uncovered by the edges of I. It is well known that if $\mathcal{F}$ is uncrossable, then so is $\mathcal{F}_I$ for any I; cf. [16].

DEFINITION 1.2. *A set $C \in \mathcal{F}$ is an $\mathcal{F}$-*core*, or simply a* core *if $\mathcal{F}$ is understood, if C does not contain two disjoint members of $\mathcal{F}$. An inclusion-minimal (inclusion-maximal) $\mathcal{F}$-core is a* min-$\mathcal{F}$-core *(*max-$\mathcal{F}$-core*). Let $\mathcal{C}(\mathcal{F})$ denote the family of min-$\mathcal{F}$-cores.*

It is not hard to verify (see Fact 2.2) that, for an uncrossable $\mathcal{F}$, if $X \in \mathcal{F}$ intersects a min-core $C \in \mathcal{C}(\mathcal{F})$, then $C \subseteq X$. In particular, the members of $\mathcal{C}(\mathcal{F})$ are pairwise disjoint, and every core X contains a unique min-core, is disjoint from all other min-cores, and is contained in a unique max-core. For $s \in V$ and $C \in \mathcal{C}(\mathcal{F})$, let

$$\mathcal{F}(s, C) = \{X : X \text{ is an } \mathcal{F}\text{-core}, X \supseteq C, s \notin X\}$$

be the family of $\mathcal{F}$-cores containing the min-core C and not containing s. A set-family is a *ring-family* if it has a unique inclusion-minimal set, and both $X \cap Y$ and $X \cup Y$ belong to the family if X, Y belong to the family. Note that $\mathcal{F}(s, C)$ is a ring-family (if $\mathcal{F}$ is uncrossable). For edge-weights, the problem of finding a minimum-weight cover of a ring-family admits a polynomial time algorithm (under Assumption 1); cf. [5]. This easily implies a 2-approximation algorithm for the node-weighted version; see section 3.

Make the following two assumptions.

Assumption 1. For any edge-set I on V, the family $\mathcal{C}(\mathcal{F}_I)$ of min-$\mathcal{F}_I$-cores can be computed in polynomial time.

Assumption 2. For any edge-set I on V, given an edge-set E on V, $s \in V$, and a min-$\mathcal{F}_I$-core C, the problem of finding a minimum node-weight $\mathcal{F}_I(s, C)$-cover contained in E admits an α-approximation algorithm.

In our applications Assumption 1 is implemented using the Ford–Fulkerson max-flow algorithm. In section 3 we show that for uncrossable $\mathcal{F}$, Assumption 1 implies Assumption 2 with $\alpha = 2$; however, in some specific applications, we might have lower values of α.

THEOREM 1.4. NWSFC *with uncrossable $\mathcal{F}$ admits a $3\alpha H(|\mathcal{C}(\mathcal{F})|)$-approximation algorithm under Assumptions* 1 *and* 2 *(here $1 \le \alpha \le 2$ is the parameter in Assumption* 2*).*

Remark. A set-function f on V is *weakly supermodular* if $f(X) + f(Y) \le f(X \cap Y) + f(X \cup Y)$ or $f(X) + f(Y) \le f(X - Y) + f(Y - X)$ for any $X, Y \subseteq V$. An edge-set I covers f if $\deg_I(X) \ge f(X)$ for all $X \subset V$. Uncrossable families correspond to $0, 1$-valued weakly supermodular set functions. For *edge-weights*, the 2-approximation of [30] for uncrossable set-families was extended to arbitrary weakly supermodular set-functions by Jain [16]. A natural question is whether Theorem 1.4 extends to weakly supermodular set-functions. As NWkF is a particular case of the problem of finding a minimum node-weight edge-cover of a weakly supermodular set-function, such an extension is unlikely even for set functions arising from NWkF, due to our hardness result given in Theorem 1.2. However, under certain assumptions, Theorem 1.4 can be used to derive a $6f_{\max} \cdot H(|V|)$-approximation algorithm for the problem of finding a minimum node-weight edge-cover of a weakly supermodular set-function, where $f_{\max} = \max_{X \subset V} f(X)$.

The main tool used to prove Theorem 1.4 is a novel decomposition of edge-covers of uncrossable families, generalizing the Klein–Ravi [18] decomposition of a forest into spiders. As uncrossable families and spiders arise in various network design problems (cf. [18, 14, 3, 24, 28, 21]), we believe that our decomposition can have further applications (e.g., to extend these algorithms from 0, 1-requirements to more general requirements). However, even extending properly the notions of "spider" and "spider-decomposition" to set-families is already a nontrivial task. Unlike the authors of [18], we cannot use graph properties to define and prove our decomposition, but can rely only on properties of uncrossable families. To prove that such a decomposition exists we use the method of laminar witness families [30, 11, 13], some ideas from [24], and some new techniques. Note also that for NWSFC our ratio is $3\alpha H(n)$ and *not* $2\alpha H(n)$ as in [18]; for a reason for that, see Lemma 3.4. In addition, $\alpha = 1$ in [18], while we could establish only $\alpha = 2$ in our more general setting.

This paper is organized as follows. Section 2 presents our main tool—a novel decomposition of covers of uncrossable families. Applications—Theorems 1.4, 1.1, and 1.3—are proved in sections 3, 4, and 5, respectively. The hardness of approximation result—Theorem 1.2—is proved in section 6.

2. Decomposition of covers of uncrossable families.

2.1. Spider-covers and decompositions. We start by describing the decomposition of [18] of a tree (or of a forest) into spiders.

DEFINITION 2.1. *A* spider *is a tree with at least two leaves and with at most one node of degree* ≥ 3. *A* spider decomposition $\mathcal{S}$ *of a tree* T *is a collection of node-disjoint spiders, each of them a subgraph of* T*, such that every leaf of* T *belongs to exactly one spider of* $\mathcal{S}$.

LEMMA 2.1 (see [18]). *Any tree* T *with at least two nodes admits a spider decomposition.*

Proof. Root T at an arbitrary leaf r. Proceed by induction on the number ℓ of leaves in T distinct from r. If $\ell = 1$ (T is a path), the statement is trivial. Otherwise, T has a node s of degree ≥ 3 so that the subtree S that consists of s and all its descendants is a spider with at least two leaves. If T is not a spider, s has an ancestor s' so that the degree of s' is at least 3, but every node in the (possibly empty) set P of the internal nodes of the $s's$-path in T has degree 2. Let $T' = T - (S \cup P)$. Note that s' is *not* a leaf of T'; hence the sets of leaves of T' and S partition the set of leaves of T. By the induction hypothesis, T' admits a spider-decomposition $\mathcal{S}'$. Thus $\mathcal{S}' \cup \{S\}$ is a spider-decomposition of T. $\square$

In this section we suggest our analogue of spiders for covers of set-families and state our main result. We need some definitions and simple facts. We will often use the following property of cores.

FACT 2.2. *Let* $\mathcal{F}$ *be an uncrossable family. If* $X \in \mathcal{F}$ *intersects a min-core* $C \in \mathcal{C}(\mathcal{F})$*, then* $C \subseteq X$*. Thus the min-*$\mathcal{F}$*-cores are pairwise disjoint, and every* $\mathcal{F}$*-core contains a unique min-*$\mathcal{F}$*-core and is disjoint from all other min-cores. Furthermore, for any two cores* X, Y*,*

- $X \cap Y, X \cup Y \in \mathcal{F}$ *if and only if* X, Y *contain the same min-core;*
- $X - Y, Y - X \in \mathcal{F}$ *if and only if* X, Y *contain distinct min-cores.*

Proof. Clearly, the min-cores are just the inclusion-minimal members of $\mathcal{F}$. If $X \in \mathcal{F}$ intersects a min-core $C \in \mathcal{C}(\mathcal{F})$, then $C \cap X \in \mathcal{F}$ or $C - X \in \mathcal{F}$, since $\mathcal{F}$ is

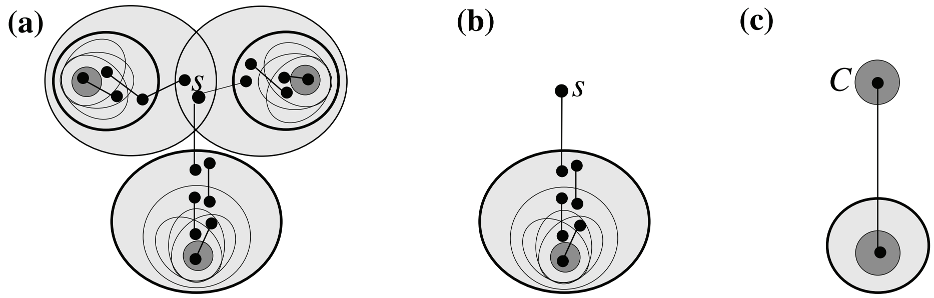

FIG. 1. *Examples of spider-covers; min-cores are shown by dark gray circles, max-cores by large light gray circles.* (a) *Note that there are several possible choices of s and that s can belong to several max-cores.* (b) *The case $\mathcal{C} = \{C\}$: all cores containing C must be covered.* (c) *An edge connecting two min-cores is also a spider-cover; then s belongs to one of the min-cores, say C, and the corresponding family $\mathcal{F}(s, C)$ and the $\mathcal{F}(s, C)$-cover P_C are both empty.*

uncrossable. As C is a min-core, we must have $C \cap X = C$ or $C - X = C$, which implies $C \subseteq X$. Hence every core contains a unique min-core and is disjoint from all other min-cores. If two cores X, Y intersect, then $X \cap Y \in \mathcal{F}$ or $X - Y, Y - X \in \mathcal{F}$. It is easy to see that the former can happen if and only if X, Y contain the same min-core, and that the latter can happen if and only if X, Y contain distinct min-cores. $\square$

Another possible proof of Lemma 2.1 is as follows. Let U be the set of leaves of T. Consider the set-family $\mathcal{F} = \{X \subset V : X \cap U \neq \emptyset, X - U \neq \emptyset\}$. It is easy to verify that $\mathcal{F}$ is uncrossable and that its set of cores is $\{X \subset V : |X \cap U| = 1\}$; the family $\mathcal{C}(\mathcal{F})$ consists of singletons in U. It can be shown that any inclusion-minimal cover $I \subseteq T$ of the family of $\mathcal{F}$-cores is a collection $\mathcal{S}$ of pairwise node-disjoint spiders; consequently, $\mathcal{S}$ is a spider decomposition of T. Note that a spider with a node s of degree ≥ 3 and leaf set U' covers all $\mathcal{F}$-cores (in fact, all members of $\mathcal{F}$) that contain a node from U' and do not contain s. Motivated by the latter observation, we suggest the following analogue of spiders for covers of set-families.

DEFINITION 2.2. *Let $\mathcal{F}$ be an uncrossable set-family on V and let $\mathcal{C} \subseteq \mathcal{C}(\mathcal{F})$. An edge-set S on V is an* $\mathcal{F}(s, \mathcal{C})$-spider-cover *if (see Figures* 1 *and* 2*) $s \in V(S)$ and if the following hold:*

- *S can be partitioned into $\mathcal{F}(s, C)$-covers $\{P_C : C \in \mathcal{C}\}$ such that the node-sets $\{V(P_C) - \{s\} : C \in \mathcal{C}\}$ are pairwise disjoint;*
- *if $|\mathcal{C}| = 1$, say $\mathcal{C} = \{C\}$, then s does not belong to any $\mathcal{F}$-core containing C.*

We say that S is an $\mathcal{F}(\mathcal{C})$-spider-cover *if there exists s so that S is an $\mathcal{F}(s, \mathcal{C})$-spider-cover, and we call any such s a* center *of S. We will sometimes just say that S is a* spider-cover *if $\mathcal{C}$ is clear from the context.*

Equivalently, for $|\mathcal{C}| \geq 2$, an $\mathcal{F}(\mathcal{C})$-spider-cover S with center $s \in V(S)$ is a union of $\mathcal{F}(s, C)$-covers $\{P_C : C \in \mathcal{C}\}$ so that only s can be a common end-node of two of them. For $\mathcal{C} = \{C\}$, S is an $\mathcal{F}(\mathcal{C})$-spider-cover if and only if S covers all cores containing C; the center s of S can be chosen as an appropriate end-node of any edge covering the max-core containing C. Note that there might be (at most one) $C \in \mathcal{C}$ so that P_C does not cover C. This may happen if $|\mathcal{C}| \geq 2$ and $s \in C$ for some $C \in \mathcal{C}$; then $\mathcal{F}(s, C) = \emptyset$ and $P_C = \emptyset$ is an $\mathcal{F}(s, C)$-cover, although no edge in P_C covers C itself. Finally, note that spider-covers are much more complex objects than spiders used in [18]; e.g., they are not even connected graphs.

Our definition of "spider-cover decomposition" of covers of set-families is the following.

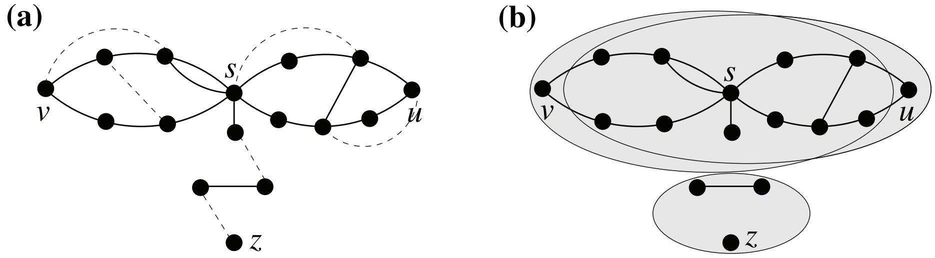

FIG. 2. *Example of an uncrossable family and a spider-cover arising from an* NWSN *instance. Let* H *be the graph formed by solid edges. The uncrossable family* $\mathcal{F} = \mathcal{F}_u \cup \mathcal{F}_v \cup \mathcal{F}_z$ *is a union of three ring-families of cuts of* H*: the family* $\mathcal{F}_u = \{X \subset V : \deg_H(X) = 2, |X \cap \{s,u\}| = 1\}$ *of cuts of size* 2 *that separate between* s *and* u*, the family* $\mathcal{F}_v = \{X \subset V : \deg_H(X) = 2, |X \cap \{s,v\}| = 1\}$ *of cuts of size* 2 *that separate between* s *and* v*, and the family* $\mathcal{F}_z = \{X \subset V : \deg_H(X) = 0, |X \cap \{s,z\}| = 0\}$ *of cuts of size* 0 *that separate between* s *and* z*.* (a) *Dashed edges form an inclusion-minimal cover of* $\mathcal{F}$*, which is also a spider-cover.* (b) *Ellipses show the max-cores of* $\mathcal{F}$ *(the min-cores are* $\{u\}, \{v\}, \{z\}$*).*

DEFINITION 2.3. *Let* I *be a cover of an uncrossable family* $\mathcal{F}$ *on* V*. A* spider-cover decomposition *of* I *is a collection of* $\mathcal{F}(\mathcal{C}_i)$*-spider-covers* $\{S_1, \dots, S_q\}$ *contained in* I *such that* $V(S_1), \dots, V(S_q)$ *are pairwise disjoint and* $\{\mathcal{C}_1, \dots, \mathcal{C}_q\}$ *is a partition of* $\mathcal{C}(\mathcal{F})$*.*

The main result of this section is the following theorem.

THEOREM 2.3 (the spider-cover decomposition theorem). *Any cover* I *of an uncrossable family* $\mathcal{F}$ *admits a spider-cover decomposition.*

Remark. In [24], a variant of Theorem 2.3 was proved for a *directed cover* of an *intersecting family*, when $X, Y \in \mathcal{F}$ and $X \cap Y \neq \emptyset$ implies $X \cap Y, X \cup Y \in \mathcal{F}$, and for every $X \in \mathcal{F}$ there should be an edge in I *entering* X. The definition of spider-cover in [24] was slightly different from the one here. For this case, in [24] it is proved that there exists a subpartition of I into $\mathcal{C}_i$-spider-covers that are pairwise *tail-disjoint*, so that the union of $\mathcal{C}_i$ contains at least $2|\mathcal{C}(\mathcal{F})|/3$ min-$\mathcal{F}$-cores (in the setting of [24], this bound is the best possible). The proof of this result is easier than that of Theorem 2.3: in the case of intersecting families, the max-cores are pairwise disjoint, and, because the edges are directed, every edge with head in some max-core can cover only cores contained in this max-core. Hence any such edge is assigned to a unique max-core. This enables us to apply some arguments as in the proof of Lemma 2.1. However, for *undirected* covers of *uncrossable* families, the situation is more involved; the max-cores may not be disjoint, many edges may cover the same max-core M, and edges contained in M may cover cores contained in other max-cores.

In what follows we prove Theorem 2.3; let $\mathcal{F}$ be an intersecting family, and let I be an *inclusion-minimal* $\mathcal{F}$-cover.

2.2. Laminar witness families. We need to establish some properties of I. By the minimality of I, for every $e \in I$ there exists $W_e \in \mathcal{F}$ such that e is the unique edge in I that covers W_e; we call such W_e a *witness set for* e; note that e might have several distinct witness sets. A family $\mathcal{W} \subseteq \mathcal{F}$ is a *witness family for* I if $\mathcal{W} = \{W_e : e \in I, W_e \text{ is a witness set for } e\}$, namely, if $|\mathcal{W}| = |I|$ and every $e \in I$ has a (unique) witness set $W_e \in \mathcal{W}$. Two sets $X, Y \subseteq V$ *cross* if each one of the sets $X \cap Y$, $X - Y$, $Y - X$ is nonempty. A set-family $\mathcal{L}$ is *laminar* if its members are pairwise noncrossing, namely, if for any intersecting $X, Y \in \mathcal{L}$ either $X \subset Y$ or $Y \subset X$ holds. Clearly, any inclusion-minimal cover I of an arbitrary set-family $\mathcal{F}$ has a witness family. The following statement has been implicitly proved in several papers; cf. [1, 30, 16].

Proposition 2.4. *Let I be an inclusion-minimal cover of an uncrossable family $\mathcal{F}$. Then there exists a witness family $\mathcal{L}$ for I that is laminar.*

Proof. By the minimality of I there exists a witness family $\mathcal{L} \subseteq \mathcal{F}$ for I. We prove that there exists such laminar $\mathcal{L}$. Among all $I' \subseteq I$ that have a laminar witness family $\mathcal{L}'$, let I' be an inclusion-maximal one. We claim that $I' = I$. Suppose to the contrary that there is $e \in I - I'$. Among all witness sets for e, let W_e be one that crosses a minimal number of sets in $\mathcal{L}'$. There is $W_f \in \mathcal{L}'$ so that W_e, W_f cross, as otherwise $\mathcal{L}' \cup \{W_e\}$ is a laminar witness family for $I' \cup \{e\}$, contradicting the maximality of I'. We claim that then at least one of the following holds:

(i) If $W_e \cap W_f, W_e \cup W_f \in \mathcal{F}$, then one of $W_e \cap W_f, W_e \cup W_f$ is a witness for one of e, f, and the other is a witness for the other.

(ii) If $W_e - W_f, W_f - W_e \in \mathcal{F}$, then one of $W_e - W_f, W_f - W_e$ is a witness for one of e, f, and the other is a witness for the other.

Consequently, at least one of the sets $W_e \cap W_f, W_e \cup W_f, W_e - W_f, W_f - W_e$ is a witness set for e. However, it is known (cf. [16]) that each of these sets crosses fewer sets in $\mathcal{L}'$ than W_e, contradicting the choice of W_e.

We now prove that (i) or (ii) must hold. Suppose that $W_e \cap W_f, W_e \cup W_f \in \mathcal{F}$. Note that then there is an edge in I covering $W_e \cap W_f$ and there is an edge in I covering $W_e \cup W_f$. However, if for arbitrary sets X, Y an edge covers one of $X \cap Y, X \cup Y$, then it covers one of X, Y, and if some edge covers both $X \cap Y$ and $X \cup Y$, then it covers both X and Y. Thus no edge in $I - \{e, f\}$ can cover $W_e \cap W_f$ or $W_e \cup W_f$, so one of e, f covers $W_e \cap W_f$, and thus the other must cover $W_e \cup W_f$. The proof of the case $W_e - W_f, W_f - W_e \in \mathcal{F}$ is similar. ∎

Let $\mathcal{L} \subseteq \mathcal{F}$ be a laminar witness family for a minimal $\mathcal{F}$-cover I. The following two simple reductions enable us to simplify the exposition.

Reduction 1. A set-family $\mathcal{F}$ is *simple* if every member of $\mathcal{F}$ is a core. It would be sufficient to prove Theorem 2.3 for simple families. This is since Definitions 2.2 and 2.3 consider covers of $\mathcal{F}$-cores only. Thus we may replace $\mathcal{F}$ by the family of $\mathcal{F}$-cores; the latter is uncrossable if $\mathcal{F}$ is, by Fact 2.2. Note that in the Node-Weighted Steiner Tree problem, $\mathcal{F} = \{X \subseteq V : X \cap U, (V - X) \cap U \neq \emptyset\}$; the spider-decomposition covers the family $\{X \subseteq V : |X \cap U| = 1\}$ of $\mathcal{F}$-cores, but may not cover the entire family $\mathcal{F}$.

Reduction 2. We may assume that the minimal members of $\mathcal{L}$ are the minimal $\mathcal{F}$-cores, namely, that $\mathcal{C}(\mathcal{L}) = \mathcal{C}(\mathcal{F})$. Otherwise (assuming $\mathcal{F}$ is simple, by Reduction 1), apply the following transformation to obtain $V', \mathcal{F}', I', \mathcal{L}'$. For every $C \in \mathcal{C}(\mathcal{F})$ do the following:

V' — add to V the new node v_C;

$\mathcal{F}'$ — replace every $X \in \mathcal{F}$ containing C by $X \cup \{v_C\}$ and add $\{v_C\}$ to $\mathcal{F}$;

I' — add to I an edge $u_C v_C$, where $u_C \in C$ arbitrary;

$\mathcal{L}'$ — replace every $X \in \mathcal{L}$ containing C by $X \cup \{v_C\}$ and add $\{v_C\}$ to $\mathcal{L}$.

This transformation is an analogue of "moving terminals to leaves" used in [18] for the Node-Weighted Steiner Tree problem. It is easy to see that $\{\{v_C\} : C \in \mathcal{C}(\mathcal{F})\}$ is the set of min-cores of both $\mathcal{F}'$ and $\mathcal{L}'$; hence $\mathcal{C}(\mathcal{F}') = \mathcal{C}(\mathcal{L}')$, as desired. It is not hard to verify that the following hold:

- The new family $\mathcal{F}'$ is simple and uncrossable if $\mathcal{F}$ is.
- I covers $\mathcal{F}$ if and only if $I' = I \cup \{u_C v_C : C \in \mathcal{C}(\mathcal{F})\}$ covers $\mathcal{F}'$.
- $\mathcal{L}'$ is a laminar witness family for I', where $\{v_C\}$ is the witness set for $u_C v_C$.
- For any $s \in V$ (so $s \neq \{v_C\}$ for every $C \in \mathcal{C}(\mathcal{F})$) and $\mathcal{C} \subseteq \mathcal{C}(\mathcal{F})$, the following holds: $S \subseteq I$ is an $\mathcal{F}(s, \mathcal{C})$-spider-cover if and only if $S' = S \cup \{u_C v_C : C \in \mathcal{C}\}$ is an $\mathcal{F}'(s, \mathcal{C}')$-spider-cover, where $\mathcal{C}' = \{\{v_C\} : C \in \mathcal{C}\}$.

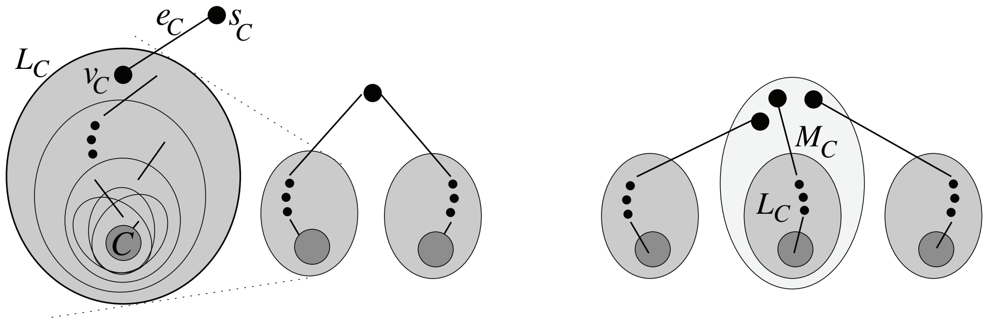

FIG. 3. *Illustration of Definition* 2.4 *(the set L_C, the edge $e_C = s_Cv_C$, and the edge-set P_C) and Definition* 2.5 *(the set M_C).*

Thus proving Theorem 2.3 for $\mathcal{F}', I'$ implies Theorem 2.3 for $\mathcal{F}, I$, provided that every spider-cover in the decomposition derived for $\mathcal{F}', I'$ has a choice of the center that belongs to V (namely, not in $\{v_C : C \in \mathcal{C}(\mathcal{F})\}$). More generally, relying on the property that there exists a laminar witness family $\mathcal{L}' \subseteq \mathcal{F}'$ for I' so that $\mathcal{C}(\mathcal{L}') = \mathcal{C}(\mathcal{F}')$, we will construct a spider-cover decomposition for I' so that every spider-cover in the decomposition has a center that does not belong to a min-core.

2.3. Proof of Theorem 2.3. In this section we finish the proof of Theorem 2.3. Assume that Reductions 1 and 2 are implemented, namely, that $\mathcal{F}$ is simple and that $\mathcal{C}(\mathcal{L}) = \mathcal{C}(\mathcal{F})$. To derive our decomposition, we will study the inclusion-maximal members of $\mathcal{L}$ and the way I covers the members of $\mathcal{F}$ contained in these sets.

DEFINITION 2.4. *For every $C \in \mathcal{C}(\mathcal{F})$ define (see Figure* 3*) the following:*

- *L_C is the maximal set in $\mathcal{L}$ containing C (L_C exists and is a core, by Reductions* 1 *and* 2*).*
- *$e_C = s_Cv_C$ is the (unique, since $L_C \in \mathcal{L}$) edge in I covering L_C, where $v_C \in L_C$.*
- *P_C is the set of edges in I with both endpoints in L_C plus e_C.*

The following statement gives some properties of the sets L_C, P_C in the above definition.

LEMMA 2.5.

(i) *The sets $\{L_C : C \in \mathcal{C}(\mathcal{F})\}$ are pairwise disjoint.*

(ii) *For every $e = uv \in I$ there is a unique $C \in \mathcal{C}(\mathcal{F})$ such that $\{u, v\} \cap L_C \neq \emptyset$; thus $P_C = \{uv \in I : \{u, v\} \cap L_C \neq \emptyset\}$, and the edge sets $\{P_C : C \in \mathcal{C}(\mathcal{F})\}$ partition I.*

(iii) *P_C covers all cores contained in L_C for every $C \in \mathcal{C}(\mathcal{F})$.*

Proof.

(i) Part (i) follows from the laminarity of $\mathcal{L}$ and the maximality of L_C.

(ii) Let $W_e \in \mathcal{L}$ be the witness set for $e \in I$. By the laminarity of $\mathcal{L}$ and the maximality of the sets L_C, $W_e \subseteq L_C$ for some $C \in \mathcal{C}(\mathcal{F})$. Consequently, e has at least one end-node in L_C. Furthermore, e has exactly one end-node in L_C if and only if $e = e_C$; in this case, L_C is the witness set for e, and thus e cannot have an end-node in $L_{C'}$ for $C' \in \mathcal{C}(\mathcal{F}) - \{C\}$, since every edge in I covers exactly one set in $\mathcal{L}$.

(iii) Part (iii) follows from part (ii) and the simple observation that if an edge e covers a set contained in L_C, then it has at least one end-node in L_C. $\square$

A family $\mathcal{L}'$ is *nested* if there exists an ordering $X_1, \ldots, X_k$ of the members of $\mathcal{L}'$ so that $X_1 \subset X_2 \subset \cdots \subset X_k$. Note that $\mathcal{L}$ is a disjoint union of nested families, whose inclusion-maximal members are the sets L_C.

By Lemma 2.5(ii), any partition $\mathcal{C}_1, \ldots, \mathcal{C}_q$ of $\mathcal{C}(\mathcal{F})$ induces the partition $S_1, \ldots, S_q$ of I, where $S_i = \cup\{P_C : C \in \mathcal{C}_i\}$. We obtain a spider-cover decomposition of I as a decomposition induced by a certain partition of $\mathcal{C}(\mathcal{F})$. Note that the edge-set $\{e_C : C \in \mathcal{C}(\mathcal{F})\}$ is a collection of pairwise node-disjoint stars, by Lemma 2.5(i). A natural partition of $\mathcal{C}(\mathcal{F})$ is by the partition induced by these stars. As we show later, every star with at least two edges induces an $\mathcal{F}(s, \mathcal{C})$-spider-cover, where s is the center of the star and $\mathcal{C}$ is the part corresponding to the edges of the star. However, this naive approach fails because for a star consisting of a single edge $e_C = s_C v_C$, the edge-set P_C may *not* cover all cores containing C, e.g., if there is a core M_C containing $L_C \cup \{s_C\}$ (see Figure 3). We will handle this difficulty by defining a partition of such "dangerous" cores, showing that every part of size at least 2 induces a spider-cover and joining every singleton part to a "nondangerous" star. This motivates the following definition.

Definition 2.5. *A min-core C is* dangerous *if there exists a core in $\mathcal{F}$ containing $L_C \cup \{s_C\}$. Let $\mathcal{D}$ denote the set of dangerous min-cores. For $C \in \mathcal{D}$, let M_C be the (unique, by Fact* 2.2*) inclusion-minimal core among the cores containing $L_C \cup \{s_C\}$.*

The following statement gives some properties of the sets M_C that we use.

Lemma 2.6. *For every $C \in \mathcal{D}$ the following hold:*

(i) *$M_C \cap L_{C'} = \emptyset$ for any $C' \in \mathcal{C}(\mathcal{F}) - \{C\}$.*

(ii) *If $e \in I$ covers M_C, then $e = e_{C'}$ for some $C' \in \mathcal{C}(\mathcal{F}) - \{C\}$ and $s_{C'} \in M_C - L_C$.*

(iii) *If $M_C \cap M_{C'} \neq \emptyset$ for $C' \in \mathcal{D}$, then $s_C, s_{C'} \in M_C \cap M_{C'}$.*

Proof.

(i) Assume to the contrary that $M_C \cap L_{C'} \neq \emptyset$ for some $C' \in \mathcal{C}(\mathcal{F}) - \{C\}$. By Fact 2.2, $M_C - L_{C'} \in \mathcal{F}$. By Lemma 2.5(i), $L_C \subseteq M_C - L_{C'}$. If $s_C \in M_C - L_{C'}$, then $L_C \cup \{s_C\} \subseteq M_C - L_{C'}$, contradicting the minimality of M_C. Otherwise, $s_C \in M_C \cap L_{C'}$; but then e_C covers $L_{C'}$, contradicting that $L_{C'}$ is a witness set for $e_{C'}$.

(ii) Let e be an edge covering M_C. By Lemma 2.5(ii) the edge-sets $\{P_C : C \in \mathcal{C}(\mathcal{F})\}$ partition I; hence $e \in P_{C'}$ for some $C' \in \mathcal{C}(\mathcal{F})$. All edges in P_C have both their end-nodes in M_C; hence $C' \neq C$. For $C' \neq C$, all edges in $P_{C'} - \{e_{C'}\}$ have both their end-nodes in $V - M_C$, by part (i) of the lemma. Part (ii) follows.

(iii) Assume to the contrary that $s_C \in M_C - M_{C'}$; the case $s_{C'} \in M_{C'} - M_C$ is identical. By Fact 2.2, $M_C - M_{C'} \in \mathcal{F}$. By part (i), $L_C \subset M_C - M_{C'}$. Hence $L_C \cup \{s_C\} \subseteq M_C - M_{C'}$, contradicting the minimality of M_C. $\square$

Corollary 2.7. *The relation $\mathcal{R} = \{(C, C') \in \mathcal{D} \times \mathcal{D} : M_C \cap M_{C'} \neq \emptyset\}$ is an equivalence.*

Proof. Clearly, $\mathcal{R}$ is symmetric and reflexive; transitivity is by Lemma 2.6(iii). $\square$

Lemma 2.8. *If $X \in \mathcal{F}$ contains $C \in \mathcal{C}(\mathcal{F})$ and is not covered by P_C, then $s_C \in X$.*

Proof. By Fact 2.2, $X \cap L_C \in \mathcal{F}$. Hence $X \cap L_C$ is covered by some edge $e \in P_C$, by Lemma 2.5(iii). Note that if an edge covers the intersection of two sets, then it covers one of the sets. This implies that e covers X or L_C, but as P_C does not cover X and $e \in P_C$, e covers L_C. One can easily verify that we must have $e = e_C$ (since e_C is the only edge in I that covers L_C), $v_c \in X \cap L_C$ (since e_C covers $X \cap L_C$), and $s_C \in X - L_C$ (since e_C does not cover X). $\square$

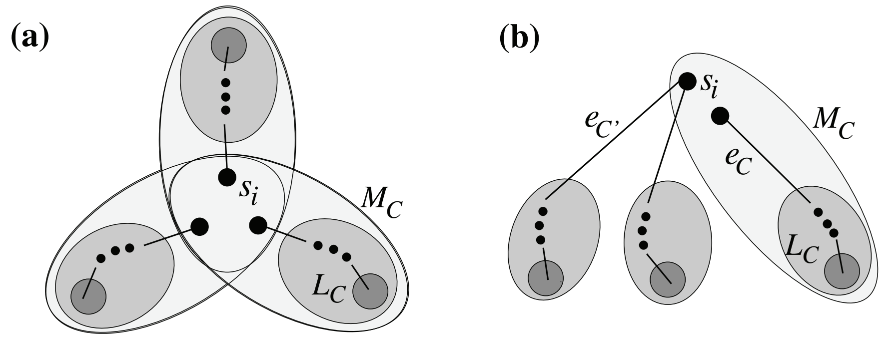

FIG. 4. (a) $\mathcal{C}_i$ *is an equivalence class of the relation* $\mathcal{R} = \{(C, C') \in \mathcal{D} \times \mathcal{D} : M_C \cap M_{C'} \neq \emptyset\}$. (b) $\mathcal{C}_i$ *is obtained from a part corresponding to a star with center* s_i *by joining a core* $C \in \mathcal{D}$ *with* $s_i \in M_C$.

COROLLARY 2.9. *If* $C \in \mathcal{C}(\mathcal{F}) - \mathcal{D}$*, then* P_C *covers any* $X \in \mathcal{F}$ *containing* C.

Proof. Suppose to the contrary that there is $X \in \mathcal{F}$ containing C that is not covered by P_C. By Lemma 2.8, $s_C \in X$. But then the set $X \cup L_C$ belongs to $\mathcal{F}$, by Fact 2.2, and contains $L_C \cup s_C$. This contradicts the assumption $C \notin \mathcal{D}$. ∎

COROLLARY 2.10. *If* $C \in \mathcal{D}$*, then* P_C *is an* $\mathcal{F}(s, C)$*-cover for any* $s \in M_C - L_C$.

Proof. Suppose to the contrary that there is $X \in \mathcal{F}(s, C)$ that is not covered by P_C. By Lemma 2.8, $s_C \in X$. By Fact 2.2, $Y = (L_C \cup X) \cap M_C \in \mathcal{F}$. Also, $L_C \cup \{s_C\} \subseteq Y$ and $Y \subseteq M_C - \{s\}$. This contradicts the minimality of M_C. ∎

Recall that, by Lemma 2.5(ii), any partition Π of $\mathcal{C}(\mathcal{F})$ induces a partition of I, where for a part $\mathcal{C} \in \Pi$ corresponds the edge-set $S = \cup\{P_C : C \in \mathcal{C}\}$. We obtain a spider-cover decomposition of I as a decomposition induced by a partition $\mathcal{C}_1, \ldots, \mathcal{C}_q$ of $\mathcal{C}(\mathcal{F})$, where for each part $\mathcal{C}_i$ we will also assign a node s_i as a center. This is done in several steps, as follows.

Let $\mathcal{A} = \{C : C \in \mathcal{D} \text{ and } \deg_I(s_C) = 1\}$. Let Π' be the subpartition of $\mathcal{A}$ into equivalence classes of size at least 2 of the relation $\mathcal{R}$ from Corollary 2.7. The node s_i assigned to a part $\mathcal{C}_i \in \Pi'$ is any s_C so that $C \in \mathcal{C}_i$ (see Figure 4(a)); thus, there are exactly $|\mathcal{C}_i|$ distinct choices of s_i, and we fix one of them. Let $\mathcal{C}'$ be the union of the parts of Π', and note that we may have $\mathcal{A} - \mathcal{C}' \neq \emptyset$ because the singleton classes of $\mathcal{R}$ are not included in Π'.

Let Π'' be a partition of $\mathcal{C}'' = \mathcal{C}(\mathcal{F}) - \mathcal{C}'$ defined as follows. First, partition $\mathcal{C}'' - \mathcal{A}$ according to stars of the graph formed by the edges e_C; namely, the parts are the equivalence classes of the relation $\{(C, C') : s_C = s_{C'}\}$ on $\mathcal{C}'' - \mathcal{A}$. A star might consist of a single edge e_C, and in this case the node s_C is the center of the star. The node s_i assigned to part $\mathcal{C}_i$ is the center of the corresponding star. Second, join every $C \in \mathcal{A} \cap \mathcal{C}''$ to some part of $\mathcal{C}'' - \mathcal{A}$ as follows (see Figure 4(b)). By Lemma 2.6(ii) and the definition of $\mathcal{R}$ and Π', there exists $C' \in \mathcal{C}'' - \mathcal{A}$ so that $e_{C'}$ covers M_C, namely, so that $s_{C'} \in M_C - L_C$; we join C to the part containing C'. Note that indeed $C' \notin \mathcal{A}$, as otherwise C, C' would belong to a part of Π'.

Let $\Pi' \cup \Pi'' = \{\mathcal{C}_1, \ldots, \mathcal{C}_q\}$ be the partition of $\mathcal{C}(\mathcal{F})$ obtained, let $\{S_1, \ldots, S_q\}$ be the partition of I induced by Π, and for each i let s_i be the center assigned to $\mathcal{C}_i$. By the construction and Lemma 2.5(ii), the node-sets $V(S_i)$ are pairwise disjoint. To finish the proof of Theorem 2.3, it is sufficient to prove that (see Definitions 2.2 and 2.3) the following lemma holds.

LEMMA 2.11. *Every* S_i *is a* $\mathcal{F}(s_i, \mathcal{C}_i)$*-spider-cover.*

Proof. By the construction, $s_i \in V(S_i)$ for all i. Also by the construction, if $\mathcal{C}_i = \{C\}$, then C is not dangerous; hence P_C covers all cores containing C in this case by Corollary 2.9. Suppose therefore that $|\mathcal{C}_i| \geq 2$. The node sets $\{V(P_C) - \{s_i\} : C \in \mathcal{C}_i\}$ are pairwise disjoint by Lemma 2.5. We claim that P_C is a $\mathcal{F}(s_i, C)$-cover for every $C \in \mathcal{C}_i$. There are two cases: $\mathcal{C}_i \in \Pi'$ and $\mathcal{C}_i \in \Pi''$.

If $\mathcal{C}_i \in \Pi'$, then s_i belongs to the intersection of the sets $\{M_C - L_C : C \in \mathcal{C}_i\}$ by the construction and Lemma 2.6. In this case P_C is a $\mathcal{F}(s_i, C)$-cover for every $C \in \mathcal{C}_i$ by Corollary 2.10.

If $\mathcal{C}_i \in \Pi''$, there are two subcases to consider: $C \in \mathcal{C}'' - \mathcal{A}$ and $C \in \mathcal{A} \cap \mathcal{C}''$. In the former subcase, our claim follows from Corollary 2.9. In the latter subcase, we have $s_i \in M_C - L_C$ by the construction and Lemma 2.6, and our claim follows from Corollary 2.10. ∎

The proof of Theorem 2.3 is complete.

3. Covering uncrossable families (proof of Theorem 1.4). We use a *greedy algorithm* for the following type of problems.

Covering Problem.
Instance: A ground-set E and integral functions ν, w on 2^E, where $\nu(E) = 0$.
Objective: Find $I \subseteq E$ with $\nu(I) = 0$ and with $w(I)$ minimized.

In the Covering Problem, the instance functions ν, w may be given by an evaluation oracle: ν is the *deficiency function* that measures how far I is from being a feasible solution, and w is the *weight function.* Let $\rho > 1$ and let opt be the optimal solution value for the Covering Problem. The *ρ-greedy algorithm* starts with $I = \emptyset$, and, as long as $\nu(I) \geq 1$, it adds to I a set $S \subseteq E - I$ so that

$$\frac{w(S)}{\nu(I) - \nu(I+S)} \leq \rho \cdot \frac{\mathsf{opt}}{\nu(I)}. \tag{2}$$

The following statement is known; we provide a simple proof of a slightly weaker statement for completeness of exposition.

Theorem 3.1. *For any* Covering Problem *so that ν is decreasing and w is increasing and subadditive, the ρ-greedy algorithm computes a solution I so that $w(I) \leq \rho H(\nu(\emptyset)) \cdot \mathsf{opt}$.*

Proof. We prove a slightly weaker result, namely, $w(I) \leq \rho(1 + \ln \nu(\emptyset)) \cdot \mathsf{opt}$. Let I_j be the partial solution at the end of iteration j, where $I_0 = \emptyset$, and let F_j be the set added at iteration j; thus $I_j = I_{j-1} \cup F_j$, $j = 1, \ldots, \ell$. Let $\nu_j = \nu(I_j)$; in particular, $\nu_0 = \nu(\emptyset)$. Since ν is decreasing, then by (2) we have

$$\frac{w(F_j)}{\nu(I_{j-1}) - \nu(I_j)} \leq \rho \cdot \frac{\mathsf{opt}}{\nu(I_{j-1})}.$$

Thus

$$\nu_j \leq \nu_{j-1} \left(1 - \frac{w(F_j)}{\rho \cdot \mathsf{opt}}\right).$$

We have $\nu_\ell = 0$, while $\nu_{\ell-1} \geq 1$. Unraveling the last inequality, we obtain

$$\frac{\nu_{\ell-1}}{\nu_0} \leq \prod_{j=1}^{\ell-1} \left(1 - \frac{w(F_j)}{\rho \cdot \mathsf{opt}}\right).$$

Taking natural logarithms from both sides and using the inequality $\ln(1+x) \le x$, we obtain

$$\rho \cdot \mathsf{opt} \cdot \ln\left(\frac{\nu_0}{\nu_{\ell-1}}\right) \ge \sum_{j=1}^{\ell-1} w(F_j).$$

Consequently, using the subadditivity of w and observing that $w(F_\ell) \le \rho \cdot \mathsf{opt}$ and $\nu_{\ell-1} \ge 1$, we get

$$w(I) = w\left(\bigcup_{j=1}^{\ell} F_j\right) \le w(F_\ell) + \sum_{j=1}^{\ell-1} w(F_j) \le \rho\cdot\mathsf{opt} + \rho\cdot\mathsf{opt}\cdot\ln\nu_0 = \rho(1+\ln\nu_0)\cdot\mathsf{opt}. \qquad \square$$

For $I \subseteq E$ define $\nu(I) = |\mathcal{C}(\mathcal{F}_I)|$ and $w(I) = w(V(I))$. Clearly, ν is decreasing, and w is increasing and subadditive. Theorem 1.4 will be proved if we prove the following lemma.

LEMMA 3.2. *For $\nu(I) = |\mathcal{C}(\mathcal{F}_I)|$ and $w(I) = w(V(I))$, an edge-set $S \subseteq E - I$ satisfying* (2) *with $\rho = 3\alpha$ can be found in polynomial time under Assumptions* 1 *and* 2.

For simplicity of exposition, let us revise our notation and use $\mathcal{F}$ instead of $\mathcal{F}_I$, and let $\nu = \nu(\emptyset)$. We assume that E is a feasible solution; thus $\nu(E) = 0$. Then we need to show that under Assumptions 1 and 2 one can find in polynomial time an edge-set $S \subseteq E$ (may not be a spider-cover) so that

$$\frac{w(S)}{\nu - \nu(S)} \le 3 \cdot \frac{\mathsf{opt}}{\nu}. \qquad (3)$$

PROPOSITION 3.3. *There exists an $\mathcal{F}(\mathcal{C})$-spider-cover S so that $w(S)/|\mathcal{C}| \le \mathsf{opt}/\nu$.*

Proof. The statement follows from Theorem 2.3 by a simple averaging argument. Let $S_1, \ldots, S_q$ be a spider-cover decomposition of an optimal $\mathcal{F}$-cover I, and let $\{\mathcal{C}_1, \ldots, \mathcal{C}_q\}$ be the corresponding partition of $\mathcal{C}(\mathcal{F})$ as in Definition 2.3. We have $\sum_{i=1}^q w(S_i) \le w(I) = \mathsf{opt}$ and $\sum_{i=1}^q |\mathcal{C}_i| = \nu$. Thus

$$\frac{\sum_{i=1}^q w(S_i)}{\sum_{i=1}^q |\mathcal{C}_i|} \le \frac{\mathsf{opt}}{\nu}\ .$$

Consequently, there must be an index i so that $w(S_i)/|\mathcal{C}_i| \le \mathsf{opt}/\nu$. $\square$

After establishing that a spider-cover satisfying $w(S_i)/|\mathcal{C}_i| \le \mathsf{opt}/\nu$ exists, we show how to find an edge-set S (may not be a spider-cover) satisfying (3) in polynomial time, under Assumptions 1 and 2. The key observation is the following.

LEMMA 3.4. *Let $\mathcal{F}$ be an uncrossable set-family on V, and let $\mathcal{C} \subseteq \mathcal{C}(\mathcal{F})$. Let S be an edge-set on V such that the following hold:*

- *If $|\mathcal{C}| \ge 2$, then there is $s \in V$ such that S is an $\mathcal{F}(s, C)$-cover for every $C \in \mathcal{C}$.*
- *If $|\mathcal{C}| = 1$, say $\mathcal{C} = \{C\}$, then S covers all $\mathcal{F}$-cores containing C.*

Then $\nu - \nu(S) \ge \max\{\lceil(|\mathcal{C}| - 1)/2\rceil, 1\} \ge |\mathcal{C}|/3$.

Proof. The min-$\mathcal{F}_S$-cores are pairwise disjoint, and each of them contains some min-$\mathcal{F}$-core. Let t be the number of min-$\mathcal{F}_S$-cores that contain exactly one min-$\mathcal{F}$-core. Any other min-$\mathcal{F}_S$-core contains at least two min-$\mathcal{F}$-cores. Thus $\nu - \nu(S) \ge \lceil(\nu - t)/2\rceil$.

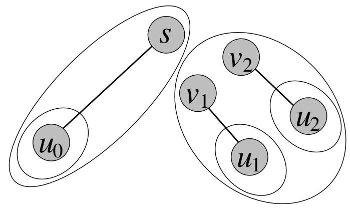

FIG. 5. *Example showing that the bound in Lemma* 3.4 *is tight for* $|\mathcal{C}| = 3$.

We upper bound t as follows. By the definition of S, any $\mathcal{F}_S$-core C' that contains some min-core $C \in \mathcal{C}$ contains s or contains some other min-$\mathcal{F}$-core distinct from C. Furthermore, if $\mathcal{C} = \{C\}$, then the latter must hold. As the min-$\mathcal{F}_S$-cores are pairwise disjoint, s belongs to at most one of them. Thus $t \leq \nu - (|\mathcal{C}| - 1)$ if $|\mathcal{C}| \geq 2$, and $t \leq \nu - 1$ if $|\mathcal{C}| = 1$. The statement follows. ∎

Remark. The bound on $\nu - \nu(S)$ given in Lemma 3.4 is tight even for laminar set-families and any $|\mathcal{C}|$; see Figure 5 for an example with $|\mathcal{C}| = 3$ and $\nu - \nu(S) = 3 - 2 = 1$. Here

$$V = \{s, u_0, u_1, u_2, v_1, v_2\},$$
$$\mathcal{F} = \{\{u_0\}, \{u_1\}, \{u_2\}, \{s, u_0\}, \{u_1, u_2, v_1, v_2\}\},$$
$$\mathcal{C} = \{\{u_0\}, \{u_1\}, \{u_2\}\}.$$

The edge-set $S = \{su_0, v_1u_1, v_2u_2\}$ is an $\mathcal{F}(s, C)$-cover for every $C \in \mathcal{C}$, and the $\mathcal{F}_S$-cores are $\{s, u_0\}$ and $\{u_1, u_2, v_1, v_2\}$. This example extends for any $|\mathcal{C}| \geq 2$. For $|\mathcal{C}| = 2k + 1$ odd, make k copies of the set $\{u_1, u_2, v_1, v_2\}$ together with the sets and edges contained in it. Then, to get an example for $|\mathcal{C}| = 2k$ even, delete u_0 together with the sets containing it and edges incident to it. This is the reason our ratio is $3\alpha H(n)$, and *not* $2\alpha H(n)$ as in [18]. One might think that a better definition of a spider-cover is an edge-set that covers *all* members of $\mathcal{F}$ separating s and some $C \in \mathcal{C}$. However, then there are examples showing that an appropriate decomposition as in Theorem 2.3 does not exist.

LEMMA 3.5. *Given* $C \in \mathcal{C}(\mathcal{F})$ *and* $v \in V$, *checking whether* v *belongs to the max-*$\mathcal{F}$*-core* M *containing* C *can be done in polynomial time under Assumption* 1.

Proof. Our decision procedure is as follows. We fix some $u \in C$. The procedure accepts v if $|\mathcal{C}(\mathcal{F}_{\{uv\}})| = |\mathcal{C}(\mathcal{F})|$. This can be checked in polynomial time by Assumption 1. Note that if $v \in M$, then the procedure accepts v, since M is disjoint from all min-$\mathcal{F}$-cores distinct from C, by Fact 2.2. Otherwise, if $v \in V - M$, then either there is no min-$\mathcal{F}_{\{uv\}}$-core that contains C (if v does not belong to any min-$\mathcal{F}$-core distinct from C) or any min-$\mathcal{F}_{\{uv\}}$-core that contains C must contain an $\mathcal{F}$-core distinct from C (if v belongs to some min-$\mathcal{F}$-core distinct from C); in both cases $|\mathcal{C}(\mathcal{F}_{\{uv\}})| = |\mathcal{C}(\mathcal{F})| - 1$, and v is rejected. ∎

Fix $v \in V$ and compute an edge-set $S_v \subseteq E$ as follows. Temporarily set the weight of v to zero. For every $C \in \mathcal{C}(\mathcal{F})$, let $W(C)$ be the weight of an $\mathcal{F}(v, C)$-cover $P(C)$ computed by the α-approximation algorithm as in Assumption 2. Sort the members of $\mathcal{C}(\mathcal{F})$ by increasing weight, say $W(C_1) \leq W(C_2) \leq \cdots \leq W(C_q)$. Let σ_j be defined as follows:

- $\sigma_1 = w(v) + \min\{W(C_i) : v \text{ is not in the max-core containing } C_i\}$ if v does not belong to some max-core, and $\sigma_1 = \infty$ otherwise.
- $\sigma_j = W_j/j$, where $W_j = w(v) + \sum_{i=1}^{j} W(C_i)$, $j = 2, \ldots, q$.

Note that $\sigma_j \leq \alpha \cdot \frac{w(S)}{j}$ for any $\mathcal{F}(v,\mathcal{C})$-spider-cover S with $|\mathcal{C}| = j$ if such exists. We find the index j for which σ_j is minimum, which determines the corresponding edge-set S_v and the set of min-cores $\mathcal{C}_v$. Specifically, if $j = 1$, then $S_v = P(C_i)$ and $\mathcal{C}_v = \{C_i\}$, where i is the index for which the minimum is attained in the definition of σ_1. If $j \geq 2$, then $S_v = \cup_{i=1}^{j} P(C_i)$ and $\mathcal{C}_v = \{C_1, \ldots, C_j\}$. Thus $\frac{w(S_v)}{|\mathcal{C}_v|} \leq \alpha \cdot \frac{w(S)}{|\mathcal{C}|}$ for any $\mathcal{F}(v,\mathcal{C})$-spider-cover S. We compute such S_v for every $v \in V$ and then, among the edge-sets $\{S_v : v \in V\}$ computed, choose one with $\frac{w(S_v)}{|\mathcal{C}_v|}$ minimum. For this choice of v we have $\frac{w(S_v)}{|\mathcal{C}_v|} \leq \alpha \cdot \frac{w(S)}{|\mathcal{C}|}$ for any $\mathcal{F}(\mathcal{C})$-spider-cover S. In particular, if S is as in Proposition 3.3, then $\frac{w(S_v)}{|\mathcal{C}_v|} \leq \alpha \cdot \frac{w(S)}{|\mathcal{C}|} \leq \alpha \cdot \frac{\mathsf{opt}}{\nu}$. On the other hand, $\frac{w(S_v)}{\nu - \nu(S_v)} \leq 3 \cdot \frac{w(S_v)}{|\mathcal{C}_v|}$, by Lemma 3.4. Consequently, $\frac{w(S_v)}{\nu - \nu(S_v)} \leq 3 \cdot \frac{w(S_v)}{|\mathcal{C}_v|} \leq 3\alpha \cdot \frac{\mathsf{opt}}{\nu}$, as required.

The time complexity is the time required to compute the family $\mathcal{C}(\mathcal{F})$ (polynomial by Assumption 1) plus $n|\mathcal{C}(\mathcal{F})|$ times the time required to check whether a given node v belongs to the max-$\mathcal{F}$-core M containing a given min-core C (polynomial by Assumption 1 and Lemma 3.5) plus $n|\mathcal{C}(\mathcal{F})|$ times the time required to find an approximately minimum-weight $\mathcal{F}(s, C)$-cover (polynomial by Assumption 2).

The proof of Lemma 3.2, and thus also of Theorem 1.4, is complete.

Finally, we will show that for uncrossable $\mathcal{F}$, Assumption 1 implies Assumption 2 with $\alpha = 2$. For that, we prove the following.

LEMMA 3.6. NWSFC *with a ring-family $\mathcal{F}$ admits a 2-approximation algorithm, provided that for any edge-set I the (unique) min-$\mathcal{F}_I$-core can be computed in polynomial time.*

Proof. We reduce the problem with a loss of a factor of 2 in the ratio to its version with edge-weights. It is well known that the edge-weighted version admits a polynomial time primal-dual/local-ratio algorithm under the assumption of the lemma. The reduction is as follows: for every edge $uv \in E$, we set its weight to be $w(uv) = \max\{w(u), w(v)\}$ and then remove the weights from the nodes. Then we compute an $\mathcal{F}$-cover $F \subseteq E$ of minimum *edge-weight*. The ratio of 2 now follows from the following known fact, which is easily deduced from Proposition 2.4 (the proof is omitted).

If F is an inclusion-minimal cover of a ring-family $\mathcal{F}$, then $\deg_F(v) \leq 2$ for all $v \in V$. ∎

4. Algorithm for NWSN (proof of Theorem 1.1). The algorithm has $r_{\max}$ iterations. Iteration k starts with a partial solution H satisfying

$$\lambda_H(u,v) \geq \min\{r(u,v), k-1\} \quad \text{for all } u,v \in V \tag{4}$$

and returns an edge-set $I \subseteq E - E(H)$ of node-weight $w(I) \leq 3H(|U|) \cdot \mathsf{opt}$ so that

$$\lambda_{H+F}(u,v) \geq \min\{r(u,v), k\} \quad \text{for all } u,v \in V. \tag{5}$$

Hence after $r_{\max}$ iterations, a feasible solution of weight at most $3r_{\max} \cdot H(|U|) \cdot \mathsf{opt}$ is found.

For $X \subseteq V$, let $r(X) = \max\{r(u,v) : |\{u,v\} \cap X| = 1\}$. By Menger's theorem, computing an augmenting edge-set I satisfying (5) is equivalent to finding a cover of the family

$$\mathcal{F} = \{X \subset V : r(X) \geq k,\ \deg_H(X) = k-1\}. \tag{6}$$

The family $\mathcal{F}$ defined by (6) is uncrossable, provided that H satisfies (4); cf. [30]. To apply Theorem 1.4, we need to show that Assumption 1 holds for $\mathcal{F}$; note that this

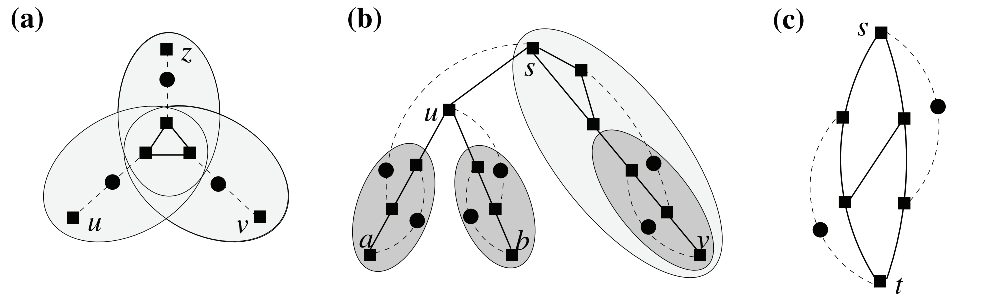

FIG. 6. *Examples of spider-covers arising from* NWSN *instances. Solid edges are already in the partial solution H, and round nodes have cost zero; dashed edges are edges of some inclusion-minimal solution. The sets L_C and M_C are shown by dark and light ellipses, respectively.* (a) *Here $k = 1$, and $r(z, u) = r(z, v) = 1$.* (b) *Here $k = 2$, and $r(s, a) = r(s, b) = r(u, v) = 2$.* (c) *Here $k = 3$, and $r(s, t) = 2$.*

implies Assumption 2 with $\alpha = 2$, by Lemma 3.6. As any edge-set I added at some previous step of iteration k can be included in H, it is sufficient to prove the following lemma.

LEMMA 4.1. *Let H satisfy* (4)*. Then Assumption* 1 *holds for $\mathcal{F}$ defined by* (6).

Proof. The minimal $\mathcal{F}$-cores can be computed using $|U|(|U|-1)/2$ max-flow computations as follows. For every pair $\{u, v\} \subseteq U$ with $r(u, v) \geq k$, compute a maximum uv-flow in H. If the flow value is $k - 1$, then in the corresponding residual directed network the set of nodes $X_{uv} = \{x \in V : x \text{ is reachable from } u\}$ is the inclusion-minimal set in $\mathcal{F}$ that contains u and does not contain v, and thus is a candidate to be the minimal core containing u; similarly, $X_{vu} = \{x \in V : v \text{ is reachable from } x\}$ is the inclusion-minimal member of $\mathcal{F}_I$ that contains v and does not contain u. The inclusion-minimal sets among all such sets, two for every pair $\{u, v\} \subseteq V$ so that $r(u, v) \geq k$ and $\lambda_H(u, v) = k - 1$, are the min-$\mathcal{F}$-cores. ∎

The proof of Theorem 1.1 is complete.

Examples. Figure 6 illustrates spider-covers arising from real NWSN instances. In the proof of Theorem 2.3 we considered two types of spider-covers in our decomposition. An example for a spider-cover formed by equivalence classes of $\mathcal{R}$ as in Figure 4(a) is given in Figure 6(a). In fact, it can be shown that for edge-connectivity, this type of spider-cover cannot occur for $k \geq 2$. However, this type of spider-cover occurs for $k = 2$ in the next section, where node-connectivity is considered, and in a related paper [27] by the author for so-called element-connectivity. The example in Figure 4(b) shows that (for edge-connectivity) spider-covers of the second type as in Figure 4(b) can occur for $k = 2$, even for laminar set-families. Finally, note that in these examples, and even in the much simpler example in Figure 6(c) of an NWkF augmentation instance, the spider-covers are not connected graphs.

5. Algorithm for $\{0, 1, 2\}$-NWSND (proof of Theorem 1.3). Let E_0 be the solution computed by the Klein–Ravi [18] (or the Guha–Khuller [14]) algorithm with the 0, 1-requirement function $\min\{r(u, v), 1\}$; then $w(V(E_0)) = O(\ln n) \cdot \mathsf{opt}$. After resetting the weight of nodes in $V(E_0)$ to 0, we get the following "residual" problem.

Instance: Disjoint edge-sets E_0, E on a node-set V, node-weights $\{w(v) : v \in V\}$ with $w(V(E_0)) = 0$, and a set D of node-pairs, so that every pair belongs to the same component of (V, E_0).

Objective: Find a minimum node-weight edge-set $I \subseteq E$ so that the graph $(V, E_0 + I)$ contains two internally disjoint uv-paths for every pair $\{u, v\} \in D$.

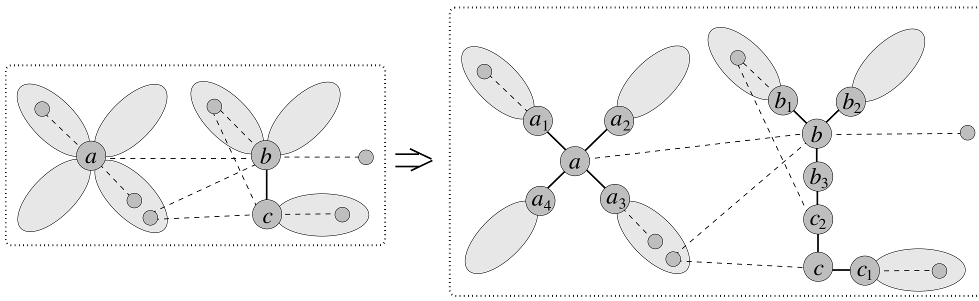

FIG. 7. *Modification of the instance $H_0 = (V, E_0), E$. Some edges in E are shown by dashed lines. The set of original cut-nodes is $Q = \{a, b, c\}$, and the added nodes are a_1, a_2, a_3, a_4, b_1, b_2, b_3, c_1, c_2.*

We reduce the latter problem to NWSFC with uncrossable $\mathcal{F}$ and show that Assumption 1 holds for this $\mathcal{F}$. We start by modifying the instance $H_0 = (V, E_0), E, w, D$ (see Figure 7). A node a is a *cut-node* of H_0 if $H_0 - a$ has more (connected) components than H_0. The components of $H_0 - a$ that are not components of H_0 are the *sides* of a. Let Q be the set of cut-nodes of H_0; if H_0 is a forest, the cut-nodes are exactly the internal (nonleaf) nodes of the trees forming H_0. For every $a \in Q$ with sides $A_1, \ldots, A_k$ do the following (see Figure 7): add new nodes $a_1, \ldots, a_k$ each of weight 0, add the edges $aa_1, \ldots, aa_k$ to E_0, and for every edge $ua \in E_0 \cup E$ with $u \in A_i$ replace its end-node a by a_i; the set D of demand pairs remains the same. Clearly, the construction is polynomial. Note that only edges in E that are incident to a node in Q and have both end-nodes in the same component of H_0 are affected. Also note that all nodes in $V(E_0)$ have weight 0. Thus for subsets of E the transformation is weight preserving, since all original nodes keep their weights, while the added nodes and the nodes in Q have weight 0. It is also easy to see that $I \subseteq E$ is a feasible solution to the original instance if and only if (the image of) I is a feasible solution to the modified instance; the node-weight of I is the same in both instances. Henceforth $H_0 = (V, E_0), E, w, D$ is the modified instance, and Q is the set of original cut-nodes. We now define our family $\mathcal{F}$ on this modified instance.

DEFINITION 5.1. *A* set-pair *is a partition $\{X, X'\}$ of $V - a$ for some $a \in Q$ so that no edge in E_0 connects X and X'. A set-pair $\{X, X'\}$ is* violated *if there is a demand pair $\{x, x'\} \in D$ so that $x \in X$ and $x' \in X'$. A set $X \subseteq V$ is violated if it is a part of some violated set-pair. Let $\mathcal{F}^+$ be the family of all violated sets, let $\mathcal{F}^- = \{V - X : X \in \mathcal{F}^+\}$, and let $\mathcal{F} = \mathcal{F}^+ \cup \mathcal{F}^-$.*

Note that $X \in \mathcal{F}^+$ if and only if $V - X \in \mathcal{F}^-$. Thus $\mathcal{F}$ is *symmetric;* that is, $X \in \mathcal{F}$ implies $V - X \in \mathcal{F}$. It is routine to show that $I \subseteq E$ is a feasible solution for the modified instance if and only if I covers $\mathcal{F}$. Lemmas 5.1 and 5.2 to follow, together with Theorem 1.4, imply Theorem 1.3.

LEMMA 5.1. *The family $\mathcal{F}$ in Definition* 5.1 *is uncrossable.*

Proof. Let $X, Y \in \mathcal{F}$. We need to show that $X \cap Y, X \cup Y \in \mathcal{F}$ or $X - Y, Y - X \in \mathcal{F}$. The following assumptions simplify the case analysis. Note that $\mathcal{F}$ is symmetric and closed under complement. Thus we may assume that the sets $X \cap Y, X - Y, Y - X$, $V - (X \cup Y)$ are all nonempty, as otherwise the statement is trivial. Also, it is enough to consider the case $X, Y \in \mathcal{F}^+$, since the sets in $\mathcal{F}^-$ are complements of the sets in $\mathcal{F}^+$, and since $\mathcal{F}$ is symmetric.

Let $\{X, X'\}$, a, $\{x, x'\}$ be as in Definition 5.1, and let $\{Y, Y'\}$, b, $\{y, y'\}$ be similarly defined. Let $A_1 = X \cap Y$, $A_2 = X \cap Y'$, $A_3 = Y' \cap X'$, and $A_4 = X' \cap Y$

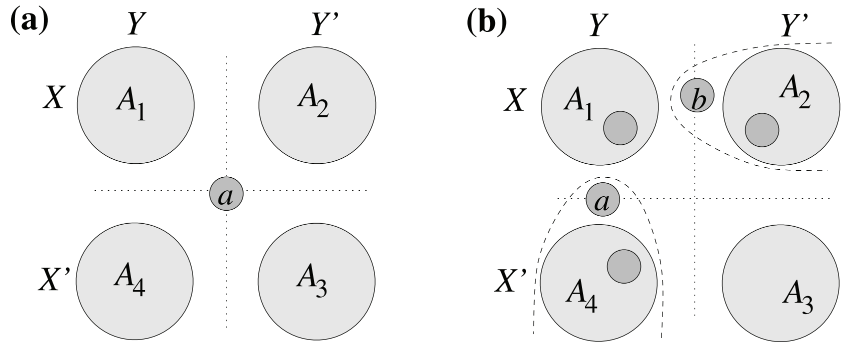

FIG. 8. *Illustration of the proof of Lemma* 5.1.

(see Figure 8). An ordered pair (s,t) of nodes is an (A_i,A_j)-pair if $s \in A_i$ and $t \in A_j$. We split the proof into two cases: $a=b$ and $a \neq b$.

Assume that $a=b$ (see Figure 8(a)). Then at least one of the following holds:

(i) (x,x') or (y,y') is an (A_1,A_3)-pair;

(ii) (x,x') or (y,y') is an (A_2,A_4)-pair;

(iii) (x,y') is an (A_1,A_3)-pair and $x',y \in A_4$, or (y,x') is an (A_1,A_3)-pair and $x,y' \in A_2$;

(iv) (x,y) is an (A_2,A_4)-pair and $x',y' \in A_3$, or (y',x') is an (A_2,A_4)-pair and $x,y \in A_1$.

One can easily verify that if (i) or (iii) holds, then $X \cap Y, X \cup Y \in \mathcal{F}^+$, and if (ii) or (iv) holds, then $X-Y, Y-X \in \mathcal{F}^+$. Hence the statement is true if $a=b$.

Assume that $a \neq b$. Then we must have $a \in Y \cup Y'$ and $b \in X \cup X'$, say $a \in Y$ and $b \in X$ (see Figure 8(b)). Note that in H_0 there is no edge between any two of the sets A_1,A_2,A_3,A_4, so any path between any two of them, if any, goes through a and/or b. In particular, x,a,x' belong to the same component of H_0; y,b,y' belong to the same component of H_0; and A_3 does not intersect the component containing a or b. Consequently, none of x,x',y,y' belongs to A_3. Eliminating from cases (i)–(iv) all the cases when one of x,x',y,y' belongs to A_3, we get that either case (ii) holds, namely, one of $(x,x'),(y,y')$ is an (A_2,A_4)-pair, or (y',x') is an (A_2,A_4)-pair and $x,y \in A_1$. In both cases, $A_2,A_4 \in \mathcal{F}^+$. The corresponding violated set-pairs are $\{A_2, V-b-A_2\}$ and $\{A_4, V-a-A_4\}$, so $V-b-A_2, V-a-A_4 \in \mathcal{F}^+$. However, the complement of $V-b-A_2$ is $A_2 \cup \{b\} = X-Y$, and the complement of $V-a-A_4$ is $A_4 \cup \{a\} = Y-X$. Hence $X-Y, Y-X \in \mathcal{F}^- \subseteq \mathcal{F}$. ∎

LEMMA 5.2. *Assumption* 1 *holds for* $\mathcal{F}$ *in Definition* 5.1.

Proof. Without loss of generality we may consider the case $I=\emptyset$. The family $\mathcal{C}(\mathcal{F})$ can be computed as follows. For every $a \in Q$ and for each side A of a we check whether A or $A \cup \{a\}$ is a violated set. Among the violated sets found, we output the inclusion-minimal ones. ∎

The proof of Theorem 1.3 is complete.

6. Hardness of NWkF (proof of Theorem 1.2). It is easy to see that NWkF is "Set-Cover hard." Indeed, the Set-Cover problem can be formulated as follows. Given a bipartite graph $J=(A+B,E)$, find minimum size subset $S \subseteq A$ such that every node in B has a neighbor in S. Construct an instance of NWkF by adding new nodes $\{s,t\}$ and edges $\{sa : a \in A\} \cup \{bt : b \in B\}$, and setting $w(v)=1$ if $v \in A$ and

$w(v) = 0$ otherwise. Then replace every edge not incident to t by $|B|$ parallel edges. For $k = |B|$, it is easy to see that S is a solution to the Set-Cover instance if and only if the subgraph induced by $S \cup B \cup \{s,t\}$ is a feasible solution to the obtained NWkF instance.

We now prove that the existence of a ρ-approximation algorithm for NWkF implies the existence of a $1/(2\rho^2)$-approximation algorithm for bipartite Densest ℓ-Subgraph. We need the following statement.

LEMMA 6.1. *There exists a polynomial time algorithm that, given a graph* $G = (V,E)$ *and an integer* $1 \le \ell \le n = |V|$*, finds a subgraph* $G' = (V',E')$ *of* G *such that* $|V'| = \ell$ *and* $|E'| \ge |E| \cdot \frac{\ell(\ell-1)}{n(n-1)}$.

Proof. While G has more than ℓ nodes, repeatedly delete the minimum-degree node from G. At the beginning of iteration $i+1$, G has $n_i = n - i$ nodes and m_i edges, where $n_0 = n$ and $m_0 = m$. The average degree is $2m_i/n_i$; thus after iteration $i+1$ the number m_{i+1} of edges in G is at least

$$m_{i+1} \ge m_i - \frac{2m_i}{n_i} = m_i \cdot \frac{n-i-2}{n-i}.$$

The statement follows since the above recursive formula implies that after $i = n - \ell$ iterations

$$\frac{m_i}{m} \ge \frac{(n-2)\cdots(n-i+1)(n-i)(n-i-1)}{n(n-1)(n-2)\cdots(n-i+1)} = \frac{(n-i)(n-i-1)}{n(n-1)} = \frac{\ell(\ell-1)}{n(n-1)}. \qquad \square$$

Given an instance $J = (A+B,E)$ and ℓ of bipartite Densest ℓ-Subgraph, define an instance of unit-weight NWkF by adding new nodes $\{s,t\}$ and edges $\{sa : a \in A\} \cup \{bt : b \in B\}$ of multiplicity $|A| + |B|$ each, and setting $w(v) = 1$ for all $v \in A \cup B$. Note that any edge-set $I \subseteq E$ determines $|I|$ edge-disjoint st-paths. Thus for any integer $k \in \{1,\ldots,|E|\}$ we have a ρ-approximation algorithm for

$$\min\{|X| : X \subseteq A + B, |E(X)| \ge k\}.$$

We show that this implies a $1/(2\rho^2)$-approximation algorithm for the Densest ℓ-Subgraph problem, which is

$$\max\{|E(X)| : X \subseteq A + B, |X| \le \ell\}.$$

For every $k = 1,\ldots,|E|$, use the ρ-approximation algorithm for NWkF to compute a subset $X_k \subseteq A + B$ so that $|E(X_k)| \ge k$, or to determine that no such X_k exists. Now, let $X = X_k$, where k is the largest integer so that $|X_k| \le \min\{\lfloor \rho \cdot \ell \rfloor, |A| + |B|\}$ and $|E(X_k)| \ge k$. Let X^* be an optimal solution for Densest ℓ-Subgraph. Note that $|E(X)| \ge |E(X^*)|$ and that $\frac{\ell(\ell-1)}{|X|(|X|-1)} \ge 1/(2\rho^2)$. By Lemma 6.1 we can find in polynomial time $X' \subseteq X$ so that $|X'| = \ell$ and

$$|E(X')| \ge |E(X)| \cdot \frac{\ell(\ell-1)}{|X|(|X|-1)} \ge |E(X^*)| \cdot \frac{1}{(2\rho^2)}.$$

Thus X' is a $1/(2\rho^2)$-approximation for the bipartite Densest ℓ-Subgraph.

The proof of Theorem 1.2 is complete.

Finally, we will give a polynomial time algorithm for the following "augmentation version" of NWkF.

Node-Weighted k-Flow Augmentation (NWkFA).
Instance: A graph $G = (V, E)$ with node weights $\{w(v) : v \in V\}$, $s, t \in V$, an integer k, and a subgraph $G_0 = (V, E_0)$ of G so that $\lambda_{G_0}(s, t) = k - 1$ and $w(V(E_0)) = 0$.
Objective: Find $F \subseteq E - E_0$ so that $\lambda_{G_0+F}(s, t) = k$ and $w(V(F))$ is minimum.

PROPOSITION 6.2. NWkFA *can be solved using one shortest-path computation.*

Proof. It would be convenient to describe the algorithm using "mixed" graphs that contain both directed and undirected edges. Given such a mixed graph with weights on the nodes, the problem of finding a minimum-weight st-path can be reduced to its edge-weighted version in a directed graph by elementary constructions (replacing every undirected edge by two opposite directed edges and converting node-weights to edge-weights). The following algorithm computes an optimal solution to NWkFA.

1. Let I_0 be an inclusion-minimal edge-set in G_0 that contains $k - 1$ pairwise edge-disjoint st-paths. Construct a mixed graph D by directing these paths from t to s.
2. Compute a minimum node-weight st-path P in D. Return $P - E_0$.

We now explain why the algorithm is correct. Let D_0 be the set of directed edges in D corresponding to I_0. From the correctness of the Ford–Fulkerson algorithm for augmenting a $(k - 1)$-flow to a k-flow and the flow decomposition theorem (cf. [5]), we have the following:

For $F \subseteq E - I_0$, $\lambda_{I_0+F}(s, t) \geq k$ if and only if $(V, D_0 + F)$ contains an st-path P.

Note that since $w(V(E_0)) = 0$, only edges in $P - E_0$ contribute to the node-weight of P. Thus NWkFA is equivalent to computing a minimum node-weight st-path P in D. As this can be implemented using one shortest-path computation, the statement follows. $\square$

7. Open problems.

We suggest the following open problems:

- Does NWSN admit an approximation ratio sublinear in $r_{\max}$? Even for NWkF, the best known ratio is k, while the reduction to Densest ℓ-Subgraph in Theorem 1.2 shows only an approximation threshold of $|V|^{1/8}$, unless for Densest ℓ-Subgraph a better algorithm than the one in [2] can be found.
- In this paper we derived a $6H(|U|) \cdot r_{\max}$-approximation algorithm for NWSN. Can the constant 6 be improved?
- Does NWSFC with ring-family $\mathcal{F}$ admit a polynomial time algorithm (under appropriate assumptions)?

Note that we recently extended the results of this paper from edge-connectivity to so-called element-connectivity; see [27].

Acknowledgments. We thank three anonymous referees for many useful comments. We also thank Danny Segev for some comments.

REFERENCES

[1] A. AGRAWAL, P. KLEIN, AND R. RAVI, *When trees collide: An approximation algorithm for the generalized Steiner problem on networks*, SIAM J. Comput., 24 (1995), pp. 440–456.
[2] A. BHASKARA, M. CHARIKAR, E. CHLAMTAC, U. FEIGE, AND A. VIJAYARAGHAVAN, *Detecting high log-densities—An $O(n^1/4)$ approximation for densest k-subgraph*, in Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC), 2010, to appear.

[3] G. Calinescu, S. Kapoor, A. Olshevsky, and A. Zelikovsky, *Network lifetime and power assignment in ad hoc wireless network*, in Proceedings of the 11th European Symposium on Algorithms (ESA), Lecture Notes in Comput. Sci. 2832, Springer, Berlin, 2003, pp. 114–126.

[4] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour, *Approximation algorithms for node-weighted buy-at-bulk network design*, in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007, pp. 1265–1274.

[5] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Scrijver, *Combinatorial Optimization*, John Wiley & Sons, New York, 1998.

[6] Y. Dodis and S. Khanna, *Design networks with bounded pairwise distance*, in in Proceedings of the 31st ACM Symposium on Theory of Computing (STOC), 1999, pp. 750–759.

[7] U. Feige, *A threshold of* $\ln n$ *for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.

[8] U. Feige, G. Kortsarz, and D. Peleg, *The dense* k*-subgraph problem*, Algorithmica, 29 (2001), pp. 410–421.

[9] M. Feldman, G. Kortsarz, and Z. Nutov, *Improved approximating algorithms for Directed Steiner Forest*, in Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2009, pp. 922–931.

[10] L. K. Fleischer, K. Jain, and D. P. Williamson, *An iterative rounding* 2*-approximation algorithm for the element connectivity problem*, in Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS), 2001, pp. 339–347.

[11] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. B. Shmoys, É. Tardos, and D. P. Williamson, *Improved approximation algorithms for network design problems*, in Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 1994, pp. 223–232.

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

[13] M. X. Goemans and D. P. Williamson, *The primal-dual method for approximation algorithms and its application to network design problems*, in Approximation Algorithms for NP-hard Problems, D. S. Hochbaum, ed., PWS, Boston, 1995, pp. 144–191.

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

[15] M. T. Hajiaghayi, G. Kortsarz, V. S. Mirokni, and Z. Nutov, *Power optimization for connectivity problems*, Math. Programming, 110 (2007), pp. 195–208.

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

[17] S. Khuller, *Approximation algorithms for finding highly connected subgraphs*, in Approximation Algorithms for NP-hard Problems, D. S. Hochbaum, ed., PWS, Boston, 1995, pp. 236–265.

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

[19] G. Kortsarz, R. Krauthgamer, and J. R. Lee, *Hardness of approximation for vertex-connectivity network design problems*, SIAM J. Comput., 33 (2004), pp. 704–720.

[20] G. Kortsarz and Z. Nutov, *Approximating minimum cost connectivity problems*, in Handbook of Approximation Algorithms and Metaheuristics, T. F. Gonzales, ed., Chapman & Hall/CRC, Boca Raton, FL, 2007, Ch. 58.

[21] G. Kortsarz and Z. Nutov, *Approximating some network design problems with node costs*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Lecture Notes in Comput. Sci. 5687, Springer, Berlin, Heidelberg, 2009, pp. 231–243.

[22] Y. Lando and Z. Nutov, *Inapproximability of survivable networks*, Theoret. Comput. Sci., 410 (2009), pp. 2122–2125.

[23] A. Moss and Y. Rabani, *Approximation algorithms for constrained node weighted Steiner tree problems*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC), 2001, pp. 373–382.

[24] Z. Nutov, *Approximating minimum power covers of intersecting families and directed connectivity problems*, in Theoret. Comput. Sci., to appear.

[25] Z. Nutov, *Approximating Steiner networks with node weights*, in LATIN 2008, Lecture Notes in Comput. Sci. 4957, Springer, Berlin, Heidelberg, 2008, pp. 411–422.

[26] Z. Nutov, *Approximating connectivity augmentation problems*, ACM Trans. Algorithms, 6 (2009), article 5.

[27] Z. Nutov, *Approximating minimum cost connectivity problems via uncrossable bifamilies and spider-cover decompositions*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2009, pp. 417–426.

[28] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, III, *Approximation algorithms for degree-constrained minimum-cost network-design problems*, Algorithmica, 31 (2001), pp. 58–78.

[29] R. Ravi and D. P. Williamson, *An approximation algorithm for minimum-cost vertex-connectivity problems*, Algorithmica, 18 (1997), pp. 21–43.

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