

Approximating minimum cost connectivity problems via uncrossable bifamilies and spider-cover decompositions

Zeev Nutov

The Open University of Israel

nutov@openu.ac.il

Abstract— We give approximation algorithms for the Generalized Steiner Network (GSN) problem. The input consists of a graph $G = (V, E)$ with edge/node costs, a node subset $S \subseteq V$, and connectivity requirements $\{r(s, t) : s, t \in T \subseteq V\}$. The goal is to find a minimum cost subgraph H that for all $s, t \in T$ contains $r(s, t)$ pairwise edge-disjoint st -paths so that no two of them have a node in $S - \{s, t\}$ in common. Three extensively studied particular cases are: Edge-GSN ($S = \emptyset$), Node-GSN ($S = V$), and Element-GSN ($r(s, t) = 0$ whenever $s \in S$ or $t \in S$). Let $k = \max_{s, t \in T} r(s, t)$. In Rooted GSN there is $s \in T$ so that $r(u, t) = 0$ for all $u \neq s$, and in the Subset k -Connected Subgraph problem $r(s, t) = k$ for all $s, t \in T$.

For edge costs, our ratios are $O(k^2)$ for Rooted GSN and $O(k^2 \log k)$ for Subset k -Connected Subgraph. This improves the previous ratio $O(k^2 \log n)$ and settles the approximability of these problems to a constant for bounded k .

For node-cost, our ratios are:

- $O(k \log |T|)$ for Element-GSN, matching the best known ratio for Edge-GSN.
- $O(k^2 \log |T|)$ for Rooted GSN and $O(k^3 \log |T|)$ for Subset k -Connected Subgraph, improving the ratio $O(k^8 \log^2 |T|)$.
- $O(k^4 \log^2 |T|)$ for GSN; this is the first non-trivial approximation algorithm for the problem.

Keywords-Generalized Steiner Network; Approximation algorithms;

1. INTRODUCTION

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

For an edge set I on node set V let $V(I) = \bigcup_{uv \in I} \{u, v\}$ denote the set of end-nodes of the edges in I . Given node costs $\{c(v) : v \in V\}$, let $c(I) = c(V(I))$ be the *node-cost* of I . For a subset S of nodes in a graph H , Let $\lambda_H^S(s, t)$ denote the S -connectivity between s and t in H , namely, the maximum number of edge-disjoint st -paths in H so that no two of them have a node in $S - \{u, v\}$ in common. We consider the following fundamental problem on undirected graphs, that includes as a special case the problems mentioned above.

Generalized Steiner Network (GSN)

Instance: A graph $G = (V, E)$ with edge/node costs, $S \subseteq V$, and S -connectivity requirements $\{r(s, t) : s, t \in T \subseteq V\}$.

Objective: Find a minimum cost subgraph H of G so that $\lambda_H^S(s, t) \geq r(s, t)$ for all $s, t \in T$.

Extensively studied particular cases of GSN are: Edge-GSN ($S = \emptyset$), Node-GSN ($S = V$), and Element-GSN ($r(s, t) = 0$ whenever $s \in S$ or $t \in S$). Element-GSN is essentially the edge-connectivity version of the problem on hypergraphs, studied in the 90's by Frank, Benczur, and many others; see e.g. [21] and the references therein. We note that GSN can be reduced to Node-GSN by elementary constructions (so all our result for Node-GSN extend to GSN). In Rooted GSN there is $s \in T$ so that $r(u, t) = 0$ for all $u \neq s$ and in Subset k -Connected Subgraph $r(s, t) = k$ for all $s, t \in T$. The later problem generalizes the k -Connected Subgraph problem; see [23], [9] and the references therein. We refer the reader to a survey in [17], and here mention some literature relevant to this paper. Let $k = \max_{s, t \in T} r(s, t)$. The first approximation algorithms for the problem appeared in the 90's for the Steiner Forest problem – the case $k = 1$. Agrawal, Klein, & Ravi [1] gave a 2-approximation for edge costs, and Klein & Ravi [15] gave an $O(\log n)$ -approximation for node costs. The latter ratio is essentially (up to constants) the best possible, as in the node costs version is Set-Cover hard [15].

For $k \geq 2$, a line of research initiated by Frank, Goemans & Williamson, and others, was to study a more general setting of edge-covering the "set-function" arising from the GSN variant. For example, Edge-GSN can be formulated as a Set-Function Edge-Cover problem as follows. Let $\delta_H(X, Y)$ denote the set of the edges from X to Y in a graph H . By Menger's Theorem a subgraph H of G is a feasible solution to a GSN instance if, and only if $|\delta_H(X, V-X)| \geq f(X)$ for all $X \subseteq V$, where $f(X) = \max\{r(s, t) : |X \cap \{s, t\}| = 1\}$ is the set-function to be edge-covered. This set-function f is (positively) *weakly supermodular*, namely, $f(X) + f(Y) \leq \max\{f(X \cap Y) + f(X \cup Y), f(X - Y) + f(Y - X)\}$ for all $X, Y \subseteq V$ with $f(X), f(Y) > 0$. A set family \mathcal{F} is *uncrossable* if $X \cap Y, X \cup Y \in \mathcal{F}$ or $X - Y, Y - X \in \mathcal{F}$ for any $X, Y \in \mathcal{F}$. It is known that if f is weakly supermodular then the family $\{X \subseteq V : f(X) = f_{\max}\}$ is uncrossable,

where $f_{\max} = \max_{X \subseteq V} f(X)$. Thus weakly supermodular set functions are decomposed into uncrossable set families.

The seminal paper of Jain [14], and numerous papers preceding it, considered Edge-GSN with edge costs, and developed novel tools for approximating minimum cost edge-covers of several types of set functions and families. Jain [14] gave a 2-approximation algorithm for edge-covering a weakly-supermodular set-function using the iterative rounding method. Earlier, Goemans et al. [13] gave a combinatorial (primal-dual/local-ratio) 2-approximation algorithm for the special case of uncrossable set-families. The 2-approximation of Jain [14] for Edge-GSN was extended to Element-GSN by Fleisher, Jain, & Williamson [11] and by Cheriyan, Vempala, & Vetta [5], by considering *setpair functions*. A pair $(X, X^*) \subseteq V \times V$ is a *setpair* (of V) if $X \cap X^* = \emptyset$. Let

$$r(X, X^*) = \max\{r(s, t) : |X \cap \{s, t\}| = |X^* \cap \{s, t\}| = 1\}.$$

By the S -connectivity version of Menger's Theorem, a subgraph H of G is a feasible solution to a GSN instance if, and only if $|\delta_H(X, X^*)| \geq f(X, X^*)$ for all setpairs (X, X^*) , where here f is a setpair function defined by

$$f(X, X^*) = r(X, X^*) - |V - (X \cup X^*)|$$

if $V - S \subseteq X \cup X^*$, and $f(X, X^*) = 0$ otherwise. This model was introduced by Frank and Jordán [12].

Recently, a progress was achieved also for node costs. In [22] the author developed an $O(\log |T|)$ -approximation algorithm for edge-covering an uncrossable *set-family* by a minimum *node-cost* edge set. This algorithm generalizes the algorithm of Klein and Ravi [15] for GSN with $k = 1$, and for node costs implies an $O(k \log |T|)$ ratio for Edge-GSN, and also for Node-GSN with $k \leq 2$. In [22] it is also proved that for large values of k , even the simplest version of Edge-GSN when $r(s, t) \neq 0$ for only one pair s, t , is at least as hard to approximate as the Densest k -Subgraph problem.

We survey some result for Node-GSN with edge costs. A hardness result of Kortsarz et al. [16] suggests that Subset k -Connected Subgraph is unlikely to admit a polylogarithmic approximation; this is so even when the input graph is complete and the costs are in $0, 1$ [21]. Chuzhoy & Khanna [2] extended this to $\Omega(k^\varepsilon)$ -hardness for any $k \geq k_0$, where k_0 and $\varepsilon > 0$ are universal constants. Lando and the author [18] proved that for $k = n/2 + k'$ the approximability of the undirected Node-GSN variant is the same (up to factor of 2) as that of the directed one with maximum requirement k' . This is so also for Rooted Node-GSN. The directed variant of Rooted GSN includes as a special case, when $k' = 1$, the Directed Steiner Tree problem. The latter is not known to admit a polylogarithmic approximation, but admits an $O(n^\varepsilon)$ -approximation scheme [3]; for $k' = 2$ no sublinear approximation for the directed rooted variant is known. On the positive side, the best known ratios were:

$O(k^3 \log n)$ for Node-GSN and $O(k^2 \log n)$ for Subset k -Connected Subgraph by Chuzhoy & Khanna [7], [8], and $O(k^2 \log n)$ for Rooted Node-GSN by Chuzhoy & Khanna [8] and the author [20]. GSN also admits an $O(\log k)$ -approximation for metric edge costs, by Cheriyan & Vetta [6]. For node costs, non-trivial algorithms were known only for Rooted GSN; $O(k^8 \log^2 n)$ -approximation by Chuzhoy & Khanna [8]. We note that in [23] the author announced an $O(k^4 \log^2 n)$ -approximation algorithm for Rooted Node-GSN with node costs, but a full proof of this result was not published, since the much better result presented in this paper was found.

We study setpair families arising from Rooted GSN instances. For all applications considered in this paper it suffices to consider setpair families that are:

- *bijective* – $(X, X^*), (X, Y^*) \in \mathcal{F}$ implies $X^* = Y^*$;
- *monotone* – $X \subseteq Y$ implies $X^* \supseteq Y^*$.

We describe such setpair families as follows. Let \mathcal{F} be a set-family on V . Suppose that to every $X \in \mathcal{F}$ corresponds a set $X^* \subseteq V - X$, so that $X \subseteq Y$ implies $X^* \supseteq Y^*$. We call such \mathcal{F} together with the mapping $X \rightarrow X^*$ a *bifamily*; for simplicity of exposition we will say that \mathcal{F} is a bifamily, without mentioning explicitly the mapping $X \rightarrow X^*$. Let $\Gamma(X) = V - (X \cup X^*)$. For an edge-set I let $\delta_I(X)$ denote the set of edges in I with one endnode in X and the other in X^* . We say that I covers a sub-bifamily $\mathcal{F}' \subseteq \mathcal{F}$ if $|\delta_I(X)| \geq 1$ for all $X \in \mathcal{F}'$. Here we suggest the following new type of bifamilies.

Definition 1.1: Given a bifamily \mathcal{F} on V and a set $T \subseteq V$ of terminals, we say that $X, Y \in \mathcal{F}$ are *dependent* if $X \cap T \subseteq \Gamma(Y)$ or if $Y \cap T \subseteq \Gamma(X)$, and X, Y are *independent* otherwise. We say that \mathcal{F} is (T, ℓ) -*uncrossable* if $X \cap T, X^* \cap T \neq \emptyset$ and $|\Gamma(X) \cap T| \leq \ell$ for all $X \in \mathcal{F}$, and if for any independent $X, Y \in \mathcal{F}$ at least one of the following holds:

$$X \cap Y, X \cup Y \in \mathcal{F} \quad \text{and} \quad (X \cap Y)^* = X^* \cup Y^* \quad (1)$$

$$(X \cup Y)^* = X^* \cap Y^*$$

$$X \cap Y^*, Y \cap X^* \in \mathcal{F} \quad \text{and} \quad (X \cap Y^*)^* = X^* \cup Y \quad (2)$$

$$(Y \cap X^*)^* = X \cup Y^*$$

We will simply say that a bifamily is *uncrossable* if it is $(T, 0)$ -uncrossable. As we will see later, uncrossable bifamilies arise from Element-GSN instances that seek to increase the connectivity by 1 between given pairs. We consider the following generic problem which includes several GSN problems.

(T, ℓ) -Uncrossable Bifamily Edge-Cover

Instance: A graph $G = (V, E)$ with edge/node costs and a (T, ℓ) -uncrossable bifamily \mathcal{F} on V .

Objective: Find a minimum cost \mathcal{F} -cover $I \subseteq E$.

We give approximation algorithms for both edge and node costs, but their polynomial implementation requires that

certain queries related to \mathcal{F} can be answered in polynomial time. In our applications, these queries are answered using max-flow and min-cost k -flow algorithms. We need some definitions to describe these queries.

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 distinct inclusion-minimal members of \mathcal{F} . An inclusion-minimal (maximal) \mathcal{F} -core is a *min- \mathcal{F} -core* (*max- \mathcal{F} -core*). Let $\mathcal{C}_{\mathcal{F}}$ denote the family of min-cores.

Given an edge set I on V (I is a partial cover of \mathcal{F}), the *residual bifamily* \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 easy to verify that if \mathcal{F} is (T, ℓ) -uncrossable, so is \mathcal{F}_I , for any I , c.f. [11] for the case of uncrossable bifamilies (the case $\ell = 0$).

Assumption 1:

For any edge set I on V , the following can be computed in polynomial time:

- (i) The family of min- \mathcal{F}_I -cores.
- (ii) The family of max- \mathcal{F}_I -cores.

For the case of node costs, it would be convenient to introduce another assumption. For $s \in V$ and $C \in \mathcal{C}_{\mathcal{F}}$ let

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

be the family of cores X containing the min-core C so that $s \in X^*$. Note that $\mathcal{F}(s, C)$ is an intersecting bifamily (namely, $X \cap Y, X \cup Y \in \mathcal{F}$ and (1) holds whenever $X, Y \in \mathcal{F}$) that has a unique inclusion minimal set; we call such a bifamily a *ring bifamily*.

Assumption 2:

Given disjoint edge-sets E, I on V , $s \in V$, and a min- \mathcal{F}_I -core C , the problem of finding a minimum node-cost $\mathcal{F}_I(s, C)$ -cover contained in E admits an α -approximation algorithm.

Later, we will show that for uncrossable \mathcal{F} , Assumption 1(i) implies Assumption 2 with $\alpha = 2$.

Our first result is the following very simple but powerful decomposition, which is obtained by a slightly improved analysis of the result of the author from [20].

Theorem 1.1: For edge/node costs, if Uncrossable Bifamily Edge-Cover admits a ρ -approximation algorithm, then (T, ℓ) -Uncrossable Bifamily Edge-Cover admits an $O(\rho(\ell + 1))$ -approximation algorithm.

For edge costs, Uncrossable Bifamily Edge-Cover admits a 2-approximation algorithm [11], [5], that uses the iterative rounding method, and applies for a more general set-function edge-cover problem. A combinatorial algorithm that relies on Assumption 1(i) only can be found in [20]. For node costs, Uncrossable Bifamily Edge-Cover includes the Set-Cover problem, and thus is $\Omega(\log n)$ -hard to approximate. The only approximation algorithm known was for the case $X^* = V - X$; in [22] is given a $3\alpha H(|T|)$ -approximation

algorithm, under Assumptions 1(i) and 2, where $H(n)$ denotes the n th Harmonic number. In this paper we prove the following generalization:

Theorem 1.2: For node costs, Uncrossable Bifamily Edge-Cover admits a $3\alpha H(|T|)$ -approximation algorithm, under Assumptions 1(i) and 2 (α is the parameter in Assumption 2).

Combining, we obtain the main result of this paper:

Theorem 1.3: (T, ℓ) -Uncrossable Bifamily Edge-Cover admits the following approximation ratios, under Assumptions 1,2:

- $O(\ell + 1)$ for edge costs;
- $O(\alpha(\ell + 1) \log |T|)$ for node costs.

We now consider some applications of Theorem 1.3. For Rooted GSN, recently two different $O(k^2 \log n)$ -approximation algorithms were suggested independently in [8] and [20]. We mention that prior to the work of [8] and [20], a randomized $k^{O(k^2)} \log^4 n$ -approximation algorithm was developed by Chakraborty, Chuzhoy, & Khanna [2], which was improved to $k^{O(k)} \log n$ by Chekuri & Korula [4]. A particularly elegant and simple approach was suggested very recently by Chuzhoy & Khanna in [7]. They showed that Rooted GSN can be decomposed into p instances of Element-GSN, where $p = p(n, k)$ is the minimum number of subsets T_1, \dots, T_p of an n -element set T , so that for every pair (t, C) with $C \subset T$, $|C| = k$, $t \in V - C$, there exists T_i with $t \in T_i$ and $C \cap T_i = \emptyset$. Chuzhoy & Khanna proved that $p = O(k^2 \log n)$. One can easily see that a factor of $\log n$ is unavoidable here even for $k = 1$. However, for $k \leq 2$ GSN admits a constant ratio [24], [11], [5], and it is also not hard to design a constant ratio algorithm for $k = 3$. All this is related to a conjecture of the author [21] that GSN with edge costs admits an approximation ratio that depends on k only. This conjecture was proved recently by the author in [19] for the special case when the input graph G is complete and the costs are in $[0, 1]$. Here we prove this conjecture for Rooted GSN and Subset k -Connected Subgraph with arbitrary costs, by deducing it from Theorem 1.3.

Theorem 1.4: GSN problems admit the following approximation ratios:

- For edge costs, $O(k^2)$ for Rooted GSN and $O(k^2 \ln k)$ for Subset k -Connected Subgraph.
- For node costs, $O(k \log |T|)$ for Element-GSN, $O(k^2 \log |T|)$ for Rooted GSN, and $O(k^3 \log |T|)$ for Subset k -Connected Subgraph; in addition, Node-GSN admits the ratio $O(k^4 \log^2 |T|)$.

This settles the approximability of Rooted GSN and of Subset k -Connected Subgraph with edge costs to a constant, and of Element-GSN, Rooted GSN, and Subset k -Connected Subgraph with node costs to $O(\log |T|)$, for bounded values of k .

Theorems 1.1, 1.2, and 1.4, are proved in Sections 2, 3, and 4, respectively.

2. PROOF OF THEOREM 1.1

Here we prove Theorem 1.1. Let \mathcal{F} be a (T, ℓ) -uncrossable bifamily on V , let $\mathcal{C}_{\mathcal{F}} = \{C_1, \dots, C_\nu\}$ be the family of min-cores, let M_i be a max-core containing C_i , and let $\mathcal{M}_{\mathcal{F}} = \{M_1, \dots, M_\nu\}$.

Claim 2.1: If $X \in \mathcal{F}$ and $C_i \in \mathcal{C}_{\mathcal{F}}$ are independent then $C_i \subseteq X$ or $C_i \cap X \cap T = \emptyset$.

Proof: Otherwise, one of the sets $C_i \cap X, C_i \cap X^*$ is in \mathcal{F} , and is strictly contained in C_i . This contradicts the minimality of C_i . ■

Claim 2.2: For any i , the set of cores containing C_i is a ring bifamily. Thus M_i is unique.

Proof: Let X, Y be cores containing C_i . We claim that (1) holds. As $X \cap Y \cap T \supseteq C_i \cap T \neq \emptyset$, X, Y are independent. Thus (1) or (2) holds. However, (2) cannot hold, as then X will contain two disjoint sets $C_i, X \cap Y^* \in \mathcal{F}$, and thus cannot be a core. Hence (1) holds, so $X \cap Y, X \cup Y \in \mathcal{F}$. Clearly, $X \cap Y$ is a core. It remains to show that $X \cup Y$ is a core. Otherwise, there is a min-core $C_j \subseteq X \cup Y$, $j \neq i$. Then $X \cap T$ or $Y \cap T$ contains a node from $C_j \cap T$, say $X \cap C_j \cap T \neq \emptyset$. Then X, C_j are independent, and thus $C_j \subseteq X$, by Claim 2.1. This contradicts that X is a core. ■

Lemma 2.3: $C_i \cap M_j \cap T = \emptyset$ if $i \neq j$; in particular, the min-cores are pairwise disjoint on T .

Proof: This follows from Claim 2.1, as $C_i \subseteq M_j$ contradicts that M_j is a core. ■

Definition 2.1: We say that $M_i, M_j \in \mathcal{M}_{\mathcal{F}}$ are *strongly independent* if both M_i, C_j are independent and M_j, C_i are independent.

Corollary 2.4: Let $M_i, M_j \in \mathcal{M}_{\mathcal{F}}$ be strongly independent. Then for any $X \subseteq M_i$ and $Y \subseteq M_j$, $X, Y \in \mathcal{F}$, (1) holds if $i = j$, and (2) holds if $i \neq j$. Thus for any $\mathcal{M} \subseteq \mathcal{M}_{\mathcal{F}}$, if the members of \mathcal{M} are pairwise strongly independent, then the bifamily $\mathcal{F}(\mathcal{M}) = \{X \in \mathcal{F} : X \subseteq M \in \mathcal{M}\}$ is uncrossable.

Proof: For $i = j$ the statement follows from Claim 2.2. If $i \neq j$, (1) cannot hold as then X will contain two disjoint sets $C_i, X \cap Y^* \in \mathcal{F}$ and thus cannot be a core. Thus (2) holds, as claimed. ■

Lemma 2.5: If $|C_i \cap T| \geq q$ for all i , then $\mathcal{M}_{\mathcal{F}}$ can be partitioned into at most $2\lceil \ell/q \rceil + 1$ parts so that the members of each part are pairwise strongly independent, and given the families $\mathcal{C}_{\mathcal{F}}, \mathcal{M}_{\mathcal{F}}$ such a partition can be found in polynomial time. Furthermore, if $q \geq \ell + 1$ then \mathcal{F} is uncrossable.

Proof: Construct an auxiliary directed graph \mathcal{J} as follows. The node set of \mathcal{J} is $\mathcal{M}_{\mathcal{F}}$. Add an arc $M_i M_j$ if $T \cap C_i \subseteq \Gamma(M_j)$. The indegree of every node in \mathcal{J} is at most $\lceil \ell/q \rceil$, by Lemma 2.3. This implies that every

subgraph of the underlying graph of \mathcal{J} has a node of degree $\leq 2\lceil \ell/q \rceil$. A graph is d -degenerate if every subgraph of it has a node of degree $\leq d$. It is known that any d -degenerate graph can be colored in polynomial time with $(d+1)$ colors. Hence \mathcal{J} is $(2\lceil \ell/q \rceil + 1)$ -colorable, and such coloring can be computed in polynomial time. Consequently, $\mathcal{M}_{\mathcal{F}}$ can be partitioned in polynomial time into $2\lceil \ell/q \rceil + 1$ independent sets. For each such independent set \mathcal{M} , the bifamily $\mathcal{F}(\mathcal{M})$ is uncrossable, by Corollary 2.4. If $q \geq \ell + 1$ then any $X, Y \in \mathcal{F}$ are independent; otherwise, there is a core C so that $|\Gamma(X) \cap T| \geq |C \cap T| \geq \ell + 1$ or $|\Gamma(Y) \cap T| \geq |C \cap T| \geq \ell + 1$, contradicting the assumption that \mathcal{F} is (T, ℓ) -uncrossable. ■

Lemma 2.6: If I covers all \mathcal{F} -cores then every min-core C' of $\mathcal{F}' = \mathcal{F}_I$ contains at least 2 distinct min- \mathcal{F} -cores. In particular, if $|C \cap T| \geq q$ for all $C \in \mathcal{C}_{\mathcal{F}}$ then $|C' \cap T| \geq 2q$.

Proof: This follows from Claim 2.1 and the definition of M_i , by observing that every min- \mathcal{F}' -core C' is a member of \mathcal{F} , and that the min-cores of \mathcal{F}' are also pairwise disjoint on T , by Lemma 2.3. ■

Summarizing, if $|C \cap T| \geq q$ for every min-core C , then the bifamily of cores can be partitioned into at most $2\lceil \ell/q \rceil + 1$ uncrossable bifamilies, by Corollary 2.4 and Lemma 2.5. If I edge-covers each of them, then $|C' \cap T| \geq 2q$ for every min-core C' of $\mathcal{F}' = \mathcal{F}_I$. We apply this procedure iteratively, until $q \geq \ell + 1$. Then the residual bifamily is uncrossable, by Lemma 2.5. As at iteration p every core contains at least $q = 2^p$ terminals, the total number of uncrossable bifamilies we cover is $1 + \sum_{p=0}^{\lceil \lg(\ell+1) \rceil} (2\lceil \ell/2^p \rceil + 1) = O(\ell + 1)$.

The proof of theorem 1.1 is complete.

3. PROOF OF THEOREM 1.2

A. Decomposition of covers of uncrossable bifamilies

In this section we extend the concept ‘‘spider-cover’’ introduced in [22] from set-families to bifamilies.

Definition 3.1: 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 $s \in V(S)$ and if the following holds:

- S can be partitioned into $\mathcal{F}(s, \mathcal{C})$ -covers $\{P_i : C_i \in \mathcal{C}\}$ such that the node sets $\{V(P_i) - \{s\} : C_i \in \mathcal{C}\}$ are pairwise disjoint;
- If $|\mathcal{C}| = 1$, say $\mathcal{C} = \{C_i\}$, then s belongs to M_i^* , where M_i is the max-core containing C_i .

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 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 $\mathcal{F}(s, C_i)$ -covers $\{P_i : C_i \in \mathcal{C}\}$ so that only s can be a common end-node of two of them. For $\mathcal{C} = \{C_i\}$, S is an $\mathcal{F}(\mathcal{C})$ -spider-cover if, and only if, S covers all cores containing C_i ; the center s of S can be

chosen as an appropriate end-node of any edge covering the max-core M_i containing C_i . Note that there might be (at most one) $C_i \in \mathcal{C}$ so that P_i does not cover C_i . This may happen if $|\mathcal{C}| \geq 2$ and $s \in C_i$ for some $C_i \in \mathcal{C}$; then $\mathcal{F}(s, C_i) = \emptyset$ and $P_i = \emptyset$ is an $\mathcal{F}(s, C_i)$ -cover, although no edge in P_i covers C_i itself.

Definition 3.2: Let I be a cover of an uncrossable bifamily \mathcal{F} on V . A subpartition $\mathcal{S} = \{S_1, \dots, S_q\}$ of I is a *spider-cover decomposition* of I if $V(S_1), \dots, V(S_q)$ are pairwise disjoint, and there exists a partition $\Pi = \{C_1, \dots, C_q\}$ of $\mathcal{C}_{\mathcal{F}}$ such that each S_t is an $\mathcal{F}(C_t)$ -spider-cover.

The main result of this section is the following generalization of the main result of [22], where the case $X^* = V - X$ was considered.

Theorem 3.1: (*The Generalized Spider-Cover Decomposition Theorem*) Any cover I of an uncrossable bifamily \mathcal{F} admits a spider-cover decomposition.

Theorem 3.1 easily implies Theorem 1.2 via the greedy method, see Section 3.2. In what follows we prove Theorem 3.1; the proof is a natural extension of the proof in [22] for the case $X^* = V - X$; we present a full proof, without relying on [22], for completeness of exposition.

Let \mathcal{F} be an *uncrossable bifamily* and let I be an *inclusion minimal \mathcal{F} -cover*. We will often use the following simple properties of sets and cores in \mathcal{F} , see [20].

Fact 3.2: In each one of the following two cases an edge e covers at least one of X, Y :

- e covers $X \cap Y$ or $X \cup Y$ and (1) holds;
- e covers $X \cap Y^*$ or $Y \cap X^*$ and (2) holds.

Fact 3.3: Let \mathcal{F} be an uncrossable bifamily and let $C \in \mathcal{C}_{\mathcal{F}}$. Then $C \subseteq X$ or $C \subseteq X^*$ for any $X \in \mathcal{F}$. Thus the min- \mathcal{F} -cores are pairwise disjoint, and every \mathcal{F} -core X contains a unique min- \mathcal{F} -core and X^* contains all the other min- \mathcal{F} -cores. Furthermore, for any two cores X, Y holds:

- $X \cap Y, X \cup Y \in \mathcal{F}$ and (1) holds if, and only if, X, Y contain the same min-core.
- $X \cap Y^*, Y \cap X^* \in \mathcal{F}$ and (2) holds if, and only if, X, Y contain distinct min-cores.

Recall that a set-family \mathcal{L} is *laminar* if for any distinct sets $X, Y \in \mathcal{L}$ either $X \subset Y$, or $Y \subset X$, or $X \cap Y = \emptyset$. The following definition extends this to bifamilies.

Definition 3.3: A bifamily $\mathcal{L} \subseteq \mathcal{F}$ is *laminar* (with respect to the mapping $X \rightarrow X^*$) if for any distinct sets $X, Y \in \mathcal{L}$ either $X \subset Y$, or $Y \subset X$, or $X \subseteq Y^*$ and $Y \subseteq X^*$ (note that the latter implies $X \cap Y = \emptyset$).

Definition 3.4: We say that an edge set I is a *fit-cover* of a bifamily $\mathcal{L} \subseteq \mathcal{F}$, or that \mathcal{L} is a *fit-bifamily* for I , if $|\mathcal{L}| = |I|$ and for every $e \in I$ there is a *fit-set* $X_e \in \mathcal{L}$ so

that $\delta_I(X_e) = \{e\}$; namely, e is the unique edge in I that covers X_e .

By the minimality of I , for every $e \in I$ there exists $X_e \in \mathcal{F}$ such that e is the unique edge in I that covers X_e . Thus there exists $\mathcal{L} \subseteq \mathcal{F}$ so that I is a fit-cover of \mathcal{L} . The following statement from [20] shows that there exists such laminar \mathcal{L} .

Lemma 3.4 ([20]): Let I be an inclusion minimal cover of an uncrossable bifamily \mathcal{F} . Among all fit-bifamilies for I contained in \mathcal{F} , let \mathcal{L} be one with $\sum_{X \in \mathcal{L}} |X|$ minimal. Then \mathcal{L} is laminar.

Let $\mathcal{L} \subseteq \mathcal{F}$ be a laminar fit-bifamily 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 3.1 for simple families. This is since Definitions 3.1 and 3.2 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 3.3.

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} .
- X^* – the mapping $X \rightarrow X^*$ is naturally redefined by keeping the sets $\Gamma(X) = V - (X \cup X^*)$ unchanged, and defining $\{v_C\}^*$ to be the complement of $\{v_C\}$.

This transformation is an analogue of “moving terminals to leaves” used in [15] for the Node-Weighted Steiner Tree problem. It is not hard to verify that:

- 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 $u_C v_C$.
- S is an $\mathcal{F}(C)$ -spider-cover if, and only if, $S' = S \cup \{u_C v_C : C \in \mathcal{C}\}$ is an $\mathcal{F}'(C')$ -spider-cover which has a choice of the center that belongs to V , where $C' = \{\{v_C\} : C \in \mathcal{C}\}$.

Thus proving Theorem 3.1 for \mathcal{F}', I' implies Theorem 3.1 for \mathcal{F}, I , provided we choose $s \in V$ as a center for every $\mathcal{F}'(C'_t)$ -spider-cover S_t in the decomposition derived; this indeed will be the case, as in the instance after the reduction, the centers of the spider-covers in our decomposition will never belong to min-cores.

Assume that Reductions 1,2 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 maximal members of \mathcal{L} .

Definition 3.5: For every $C_i \in \mathcal{C}_{\mathcal{F}}$ define:

- L_i is the maximal set in \mathcal{L} containing C_i (L_i exists and is a core, by Reductions 1,2).
- $e_i = s_i v_i$ is the unique edge in I covering L_i , where $v_i \in L_i$ and $s_i \in L_i^*$.
- P_i is the set of edges in I with at least one end-node in L_i .

The following statement gives some properties of the sets L_i, P_i in the above definition.

Lemma 3.5:

- (i) $L_i \subseteq L_j^*$ and $L_j \subseteq L_i^*$ for any $j \neq i$.
- (ii) The edge sets P_i partition I and the node sets $V(P_i) - \{s_i\}$ are pairwise disjoint.
- (iii) If $X \in \mathcal{F}$ and $X \subseteq L_i$ then P_i covers X and no edge in $I - P_i$ covers X .

Proof: Part (i) follows from the laminarity of \mathcal{L} and the maximality of L_i . Part (ii) follows from (i) and the fact that \mathcal{L} is a fit-bifamily for I . Part (iii) follows from (ii) and the observation that if an edge e covers a set contained in L_i , then e has at least one end-node in L_i . ■

Definition 3.6: A min-core C_i is *dangerous* if there is $X \in \mathcal{F}$ so that $C_i \subset X$ and X is not covered by P_i ; we say that such X is *dangerous for C_i* . Let D_i be the (unique, by Fact 3.3) inclusion minimal dangerous set for C_i , if such exists, and let \mathcal{D} be the family of all such minimal dangerous sets D_i .

Lemma 3.6: Let $X \in \mathcal{F}$ be dangerous for C_i . Then for any $j \neq i$ the set $X \cap L_j^*$ is also dangerous for C_i . Furthermore, $X \cap L_i$ is a fit-set for e_i .

Proof: By Fact 3.3, $X \cap L_j^*, X \cap L_i \in \mathcal{F}$. Consider arbitrary edges $f_j, f_i \in I$, where f_j covers $X \cap L_j^*$ and f_i covers $L_i \cap X$. By Fact 3.2, f_j covers one of L_j, X and f_i covers one of L_i, X . If f_j covers X then $f_j \notin P_i$, since X is not covered by P_i . If f_j covers L_j then $f_j = e_j \notin P_i$. Consequently, the set $X \cap L_j^*$ belongs to \mathcal{F} , contains C_i (by Lemma 3.5 (ii)), and is not covered by P_i ; thus it is dangerous for C_i . On the other hand, $f_i \in P_i$ by Lemma 3.5 (iii). Thus f_i does not cover X , so f_i covers L_i , and hence $f_i = e_i$. This implies that $X \cap L_i$ is a fit-set for e_i . ■

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

Lemma 3.7: Let $D_i \in \mathcal{D}$. Then $D_i \subseteq L_j^*$ for all $j \neq i$. Furthermore, if $e \in I$ covers D_i then $e = e_j$ for some $j \neq i$, $s_j \in D_i - L_i$, and $s_j \in D_i - D_j^*$ if D_j exists.

Proof: Part (i) follows from Lemma 3.6 and the minimality of D_i . We prove Part (ii). As e covers D_i , $e \notin P_i$. Thus $e \in P_j$ for some $j \neq i$, by Lemma 3.5 (ii). Then e has

one endnode in L_j and the other in $D_i \subseteq L_j^*$. Consequently, e covers L_j , and thus $e = e_j$; hence $s_j \in D_i$. Also, $s_j \notin L_i$ as then e would cover L_i , by Lemma 3.5 (i). Consequently, $s_j \in D_i - L_i$. If D_j exists, then $s_j \notin D_j^*$, as otherwise e_j will cover D_j , contradicting the definition of D_j . ■

Lemma 3.8: The relation $\mathcal{R} = \{(D_i, D_j) : s_j \in D_i, i \neq j\}$ on \mathcal{D} is symmetric and transitive.

Proof: We need the following Claim:

Claim: If the set $D_i \cap D_j^*$ is strictly contained in D_i , $i \neq j$, then $s_i \in D_j - D_i^*$.

Proof: By the minimality of D_i , any set in \mathcal{F} that is strictly contained in D_i is covered by some edge in P_i . By Fact 3.3, $D_i \cap D_j^* \in \mathcal{F}$. Thus there exists $e \in P_i$ covering $D_i \cap D_j^*$. By Fact 3.2, e covers D_i or e covers D_j . As $e \in P_i$, e cannot cover D_i , so e covers D_j . Hence $e = e_i$ and $s_i \in D_j - D_i^*$, by Lemma 3.7(ii) (with the roles of i, j exchanged). This proves the claim.

We prove symmetry. Suppose that $s_j \in D_i$ for $i \neq j$. Consider the set $D_i \cap D_j^*$. Note that $s_j \notin D_j^*$ (as otherwise e_j covers D_j), hence $D_i \cap D_j^* \subseteq D_i - s_j \subset D_i$, namely, $D_i \cap D_j^*$ is strictly contained in D_i . Thus s_i belongs to $D_j - D_i^* \subset D_j$, by the Claim above.

We prove transitivity. Let D_i, D_p, D_j be distinct. Suppose that $s_p \in D_i$ and $s_j \in D_p$ and we prove that $s_j \in D_i$. Since $s_j \in D_p$ then $s_p \in D_j$, by the just proved symmetry. Hence $s_p \in D_i \cap D_j$, implying $D_j \cap D_i^* \subseteq D_j - s_p \subset D_j$, namely, $D_j \cap D_i^*$ is strictly contained in D_j . Exchanging the roles of i, j in the Claim above, we obtain $s_j \in D_i - D_j^* \subset D_i$. ■

The relation \mathcal{R} may not be reflexive. This is since we may have $s_i \in L_i^* \cap (V - (D_i \cup D_i^*))$. However, if we add to \mathcal{R} all the pairs (D_i, D_i) , then \mathcal{R} becomes an equivalence relation. The following statement explains how we intend to choose the centers of the spider-covers in our decomposition.

Lemma 3.9:

- (i) If C_i is not dangerous (so D_i does not exist) then P_i is an $\mathcal{F}(s, C_i)$ -cover for any $s \in L_i^*$.
- (ii) If C_i is dangerous (so D_i exists) then P_i is an $\mathcal{F}(s, C_i)$ -cover for any $s \in V - D_i^*$.

Proof: Part (i) is obvious, and holds for any $s \in V$ as P_i covers all cores containing C_i if C_i is not dangerous. We therefore prove Part (ii). Let $s \in V - D_i^*$. Let $X \in \mathcal{F}(s, C_i)$, so $X \supseteq C_i$ and $s \in X^*$. We need to prove that X is covered by P_i . Consider the set $X \cap D_i$. By Fact 3.3, (1) holds and $X \cap D_i \in \mathcal{F}$. Note that $X \cap D_i$ is strictly contained in D_i , as we cannot have $X \supseteq D_i$ since then we would have $X \notin \mathcal{F}(s, C_i)$. Hence, by the minimality of D_i , there is $e \in I$ covering $X \cap D_i$. By Fact 3.2, e covers X or e covers D_i , but e cannot cover D_i since $e \in P_i$. Hence e covers X , so X is covered by an edge in P_i . ■

Now we can describe our decomposition. Note that by Lemma 3.5 (ii), any partition Π of $\mathcal{C}_{\mathcal{F}}$ induces a partition \mathcal{S}

of I , where for a part $\mathcal{C} \in \Pi$ corresponds the edge set part $S = \cup\{P_i : C_i \in \mathcal{C}\}$ of \mathcal{S} . We define such a partition and assign a center node to each part. In what follows, let

$$\mathcal{A} = \{C_i \in \mathcal{C}_{\mathcal{F}} : C_i \text{ is dangerous and } \deg_I(s_i) = 1\}.$$

Let Π' be the subpartition of \mathcal{A} into equivalence classes of size at least 2 of the relation \mathcal{R} from Lemma 3.8. For a part \mathcal{C} of Π' , its center s is any s_i so that $C_i \in \mathcal{C}$ (so there are $|\mathcal{C}|$ distinct choices of s , 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_i (a star might consist of a single edge); namely, the parts are the equivalence classes of the relation $\{(C_i, C_j) : s_i = s_j\}$ on $\mathcal{C}'' - \mathcal{A}$. The center of every part is the center of the corresponding star. Second, join every $C_i \in \mathcal{A} \cap \mathcal{C}''$ to some part of $\mathcal{C}'' - \mathcal{A}$ as follows. By Lemma 3.7(ii) and the definition of \mathcal{R} and Π' , there exists $C_j \in \mathcal{C}'' - \mathcal{A}$ so that e_j covers D_i , namely, so that $s_j \in D_i - L_i$; we join C_i to the part containing C_j . Note that indeed $C_j \notin \mathcal{A}$, as otherwise C_i, C_j would belong to a part of Π' .

Let $\Pi = \Pi' \cup \Pi''$ be the partition of $\mathcal{C}_{\mathcal{F}}$ obtained, and let $\mathcal{S} = \mathcal{S}' \cup \mathcal{S}''$ be the partition of I induced by Π , where \mathcal{S}' and \mathcal{S}'' are the subpartitions of I induced by Π' and Π'' , respectively. We claim that \mathcal{S} is a spider cover decomposition of I (see Definition 3.2). By the construction and Lemma 3.5(ii), the node sets $\{V(S) : S \in \mathcal{S}\}$ are pairwise disjoint. To finish the proof of Theorem 3.1, it is sufficient to prove:

Lemma 3.10: Every $S \in \mathcal{S}$ is an $\mathcal{F}(s, \mathcal{C})$ -spider-cover, for the corresponding part $\mathcal{C} \in \Pi$ and the center s chosen.

Proof: It is sufficient to show (see Definition 3.1) that the node sets $\{V(P_i) - \{s\} : C_i \in \mathcal{C}\}$ are pairwise disjoint, and that for every $C_i \in \mathcal{C}$ the following holds:

- P_i is an $\mathcal{F}(s, \mathcal{C})$ -cover;
- if $\mathcal{C} = \{C_i\}$ then P_i covers all cores containing C_i .

By the construction, if $\mathcal{C} = \{C_i\}$, then C_i is not dangerous, hence P_i covers all the cores containing C_i in this case. Suppose therefore that $|\mathcal{C}| \geq 2$. If $\mathcal{C} \in \Pi'$ then the statement follows from Lemmas 3.7 and 3.9(ii). If $\mathcal{C} \in \Pi''$, then if $C_i \notin \mathcal{A}$ then the statement follows from 3.9(i), and if $C_i \in \mathcal{A}$ then the statement follows from 3.9(ii) ■

The proof of Theorem 3.1 is complete.

B. Deducing Theorem 1.2 from Theorem 3.1

This part of the proof of Theorem 1.2 is a slight modification of the one in [22], and is stated only for completeness of exposition. We use a *Greedy Algorithm* for the following type of problems:

Covering Problem

Instance: A ground-set E and integral function ν, c on 2^E , where $\nu(E) = 0$.

Objective: Find $I \subseteq E$ with $\nu(I) = 0$ and with $c(I)$ minimized.

In the Covering Problem, the instance functions ν, c may be given by an evaluation oracle; ν is the *deficiency function* that measures how far is I from being a feasible solution, and c is the *cost 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 subset $S \subseteq E - I$ so that

$$\frac{c(S)}{\nu(I) - \nu(I \cup S)} \leq \rho \cdot \frac{\text{opt}}{\nu(I)}. \quad (3)$$

It is known (c.f. [15] for a slightly weaker statement), that for any Covering Problem so that ν is decreasing and c is increasing and sub-additive, the ρ -Greedy Algorithm computes a solution I so that $w(I) \leq \rho H(\nu(\emptyset)) \cdot \text{opt}$. For $I \subseteq E$ define: $\nu(I) = |\mathcal{C}(\mathcal{F}_I)|$, $c(I) = c(V(I))$. Clearly, ν is decreasing, and c is increasing and sub-additive. Theorem 1.2 will be proved if we prove:

Lemma 3.11: For $\nu(I) = |\mathcal{C}(\mathcal{F}_I)|$ and $c(I) = c(V(I))$, an edge set $S \subseteq E - I$ satisfying (3) with $\rho = 3\alpha$ can be found in polynomial time under Assumptions 1,2.

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

$$\frac{c(S)}{\nu - \nu(S)} \leq 3\alpha \cdot \frac{\text{opt}}{\nu}. \quad (4)$$

Proposition 3.12: There exists an $\mathcal{F}(\mathcal{C})$ -spider-cover S so that $c(S)/|\mathcal{C}| \leq \text{opt}/\nu$.

Proof: The statement follows from Theorem 3.1 by a simple averaging argument. Let S_1, \dots, S_q be a spider-cover decomposition of an optimal \mathcal{F} -cover I , and let $\{\mathcal{C}_1, \dots, \mathcal{C}_q\}$ be the corresponding partition of $\mathcal{C}_{\mathcal{F}}$ as in Definition 3.2. We have $\sum_{i=1}^q c(S_i) \leq c(I) = \text{opt}$, and $\sum_{i=1}^q |\mathcal{C}_i| = \nu$. Thus

$$\frac{\sum_{i=1}^q c(S_i)}{\sum_{i=1}^q |\mathcal{C}_i|} \leq \frac{\text{opt}}{\nu}.$$

Consequently, there must be an index t so that $c(S_t)/|\mathcal{C}_t| \leq \text{opt}/\nu$. ■

The key observation is:

Lemma 3.13: 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 holds.

- If $|\mathcal{C}| \geq 2$ then there is $s \in V$ such that S is a $\mathcal{F}(s, \mathcal{C})$ -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 $\nu - \nu(S) \geq \max\{\lceil (|\mathcal{C}| - 1)/2 \rceil, 1\} \geq |\mathcal{C}|/3$.

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 2 min- \mathcal{F} -cores. Thus $\nu - \nu(S) \geq \lceil (\nu - t)/2 \rceil$.

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 3.13 is tight even for laminar set-families and any $|\mathcal{C}|$, as was shown in [22].

Lemma 3.14: Given $C \in \mathcal{C}_{\mathcal{F}}$ and $v \in V$, checking whether $v \in M^*$ can be done in polynomial time, under Assumption 1(i), where M is the max-core containing C .

Proof: Our decision procedure is as follows. We fix some $u \in C$. The procedure accepts v if $|\mathcal{C}_{\mathcal{F}_{\{uv\}}}| \leq |\mathcal{C}_{\mathcal{F}}| - 1$. This can be checked in polynomial time, by Assumption 1(i). It is not hard to see that if $v \in M^*$, then either there is no min- $\mathcal{F}_{\{uv\}}$ -core that contains C or any min- $\mathcal{F}_{\{uv\}}$ -core that contains C must contain a min- \mathcal{F} -core distinct from C ; in both cases $|\mathcal{C}_{\mathcal{F}_{\{uv\}}}| \leq |\mathcal{C}_{\mathcal{F}}| - 1$ and v is accepted. Now suppose that $v \notin M^*$. Then, by Fact 3.3, all min- \mathcal{F} -cores distinct from C are contained in M^* , hence they are also min-cores of $\mathcal{F}_{\{uv\}}$. Also, M remains a core of \mathcal{F} . Consequently, $|\mathcal{C}_{\mathcal{F}_{\{uv\}}}| = |\mathcal{C}_{\mathcal{F}}|$ in this case, and v is rejected. ■

Fix $v \in V$ and compute an edge set $S_v \subseteq E$ as follows. Set temporarily the weight of v to zero. For every $C \in \mathcal{C}_{\mathcal{F}}$ let $W(C)$ be the node-cost 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 $W(C_1) \leq W(C_2) \leq \dots \leq W(C_q)$. Let σ_j be defined as follows:

- $\sigma_1 = w(v) + \min\{W(C_i) : v \in M_i^*\}$ if there exists an index i so that $v \in M_i^*$, and $\sigma_1 = \infty$ otherwise.
- $\sigma_j = W_j/j$ where $W_j = w(v) + \sum_{i=1}^j W(C_i)$, $j = 2, \dots, q$.

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 $\mathcal{C}_v = \{C_1, \dots, C_j\}$. 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 3.12, then $\frac{w(S_v)}{|\mathcal{C}_v|} \leq \alpha \cdot \frac{w(S)}{|\mathcal{C}|} \leq \alpha \cdot \frac{\text{opt}}{\nu}$. On the other hand, $\frac{w(S_v)}{\nu - \nu(S_v)} \leq 3 \cdot \frac{w(S_v)}{|\mathcal{C}_v|}$, by Lemma 3.13. Consequently, $\frac{w(S_v)}{\nu - \nu(S_v)} \leq 3 \cdot \frac{w(S_v)}{|\mathcal{C}_v|} \leq 3\alpha \cdot \frac{\text{opt}}{\nu}$, 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 M_i^* for a given min-core C_i (polynomial by Assumption 1 and Lemma 3.14) plus $n|\mathcal{C}_{\mathcal{F}}|$ times the time required to find a minimum-weight $\mathcal{F}(S, C)$ -cover (polynomial by Assumption 2).

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

Lemma 3.15: Ring-Bifamily Edge-Cover with node costs 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 (the proof is omitted):

If F be an inclusion-minimal cover of a ring bifamily \mathcal{F} then $\deg_F(v) \leq 2$ for all $v \in V$. ■

The proof of Lemma 3.11, and thus also of Theorem 1.2 is complete.

4. PROOF OF THEOREM 1.4

To prove Theorem 1.4, we decompose GSN into k “simple” problems of increasing the connectivity between a given set \mathcal{T} of node pairs from ℓ to $\ell + 1$, $\ell = 0, \dots, k - 1$. That is, we consider the restriction of GSN to instances when G contains an edge set J of cost 0 so that $\lambda_J^S(s, t) = \ell$ and $r(s, t) = \ell + 1$ for pairs $(s, t) \in \mathcal{T} \subseteq T \times T$, and $r(s, t) = 0$ otherwise. Formally:

Simple GSN:

Instance: A graph $G = (V, E + J)$ with edge/node costs, $S \subseteq V$, and a set $\mathcal{T} \subseteq T \times T$ of node pairs so that $\lambda_J^S(s, t) = \ell$ for all $(s, t) \in \mathcal{T}$.

Objective: Find a minimum cost edge-set $I \subseteq E$ so that $\lambda_{J+I}^S(s, t) \geq \ell + 1$ for all $\{s, t\} \in \mathcal{T}$.

Using Theorem 1.3, we will prove the following two statements, that imply Theorem 1.4.

Theorem 4.1: Simple Rooted GSN admits the the following approximation ratios: $O(k)$ for edge costs and $O(k \log |T|)$ for node costs. Rooted GSN with edge costs and with requirements in $\{0, k\}$ admits an $O(k \log k)$ -approximation algorithm.

Theorem 4.2: Simple Element-GSN with node costs admits an $O(\log |T|)$ -approximation algorithm.

Theorem 1.3 will follow from Theorems 4.1 and 4.2 by using the following known statement:

Claim 4.3: For both edge and node costs the following holds.

- (i) If Simple GSN admits a ρ -approximation algorithm then GSN admits a $k\rho$ -approximation algorithm; this is so for Rooted GSN and Subset k -Connected Subgraph.
- (ii) A ρ -approximation for Rooted GSN with requirements in $\{0, k\}$ implies a $\rho \cdot \min\{k, |T| - 1\}$ -approximation for Subset k -Connected Subgraph.

Proof: For Part (i) consider the following algorithm. Start with $J = (V, \emptyset)$ and continue with iterations. Iteration ℓ starts with a graph J with $\kappa_J(s, t) = \min\{\ell, r(s, t)\}$ for all $s, t \in T$, and seeks to increase the st -connectivity from ℓ to $\ell + 1$ for every $s, t \in T$ with $\kappa_J(s, t) = \ell$ and $r(s, t) \geq \ell + 1$; thus this is an instance of Simple GSN. We find an edge set I_ℓ of cost ρ -opt using the ρ -approximation algorithm for Simple GSN. After at most k iterations J satisfies the requirements, and its cost is $\leq k\rho \cdot \text{opt}$. The algorithm for Part (ii) is as follows. Choose arbitrary $\min\{k, |T| - 1\}$ roots and for each root s compute a ρ -approximation for Rooted GSN with requirements $r(s, t) = k$ for each $t \in T - \{s\}$. Then take the union of the $\min\{k, |T| - 1\}$ subgraphs computed. It is known and easy to see that the computed solution is feasible, and its cost is as claimed. ■

Consequently, Theorem 1.4 follows from Theorems 4.1 and 4.2, so we only need to prove the latter two theorems.

To avoid considering “mixed” cuts that contain both nodes and edges, we may assume that $st \notin E_J$ for all $\{s, t\} \in \mathcal{T}$. One way to achieve this is to subdivide every edge $st \in E_J$ with $\{s, t\} \in \mathcal{T}$ by a new node (of cost 0, in the case of node costs), and to add all these nodes to S . Let $\Gamma(X) = \{v \in V - X : uv \in E_J \text{ for some } u \in X\}$ be the set of neighbors of X in J , and let $X^* = V - (X \cup \Gamma_J(X))$. By Menger’s Theorem, I is a feasible solution to Simple GSN if, and only if, I covers the bifamily

$$\mathcal{F} = \{X \subseteq V : \exists \{s, t\} \in \mathcal{T} \text{ so that } \Gamma(X) \subseteq S, \quad (5)$$

$$|\Gamma(X)| = \ell, |X \cap \{s, t\}| = |X^* \cap \{s, t\}| = 1\}.$$

In the case of Rooted GSN, it is sufficient to cover the bifamily

$$\mathcal{F}^s = \{X \in \mathcal{F} : s \in X^*\}. \quad (6)$$

Now, we use the following statement:

Fact 4.4: For the bifamilies \mathcal{F} in (5) and \mathcal{F}^s in (6) the following holds:

- \mathcal{F} is uncrossable for Element-GSN.
- \mathcal{F}^s is (T, ℓ) -uncrossable for Rooted Node-GSN.

Proof: The fact that \mathcal{F} is uncrossable for Element-GSN is proved in [11]. The second statement is explicitly proved in [20]. ■

Using standard max-flow min-cut methods, it is easy to see that Assumption 1 holds for the family \mathcal{F} in (5), c.f. [19]. As was mentioned, Assumption 2 with parameter $\alpha = 2$ follows from Assumption 1. From this point, we will consider the cases of edge costs and node costs separately.

Edge costs: Here we consider Rooted GSN and the corresponding (T, ℓ) -uncrossable bifamily \mathcal{F}^s . For Uncrossable Bifamily Edge-Cover we have a polynomial time algorithm that computes a solution of cost at most 2 times an optimal LP-relaxation value for the problem [11], [20]. Thus from Theorem 1.3 and Fact 4.4 we obtain:

Corollary 4.5: Simple Rooted GSN admits a polynomial algorithm that computes a solution I of cost $O(\ell + 1)\tau^*$, where $\tau^* = \min\{c \cdot x : x(\delta(U)) \geq 1 \forall U \in \mathcal{F}^s, x \geq 0\}$ is the optimal LP-relaxation value for the problem.

The following widely used statement shows that the claim above implies Theorem 1.4.

Proposition 4.6: At iteration ℓ , the cost of the edge set I computed is $O((\ell + 1)/(k - \ell)) \cdot \text{opt}$. where opt denotes the optimal solution value for Rooted GSN with requirements in $\{0, k\}$.

Proof: By Corollary 4.5, it is sufficient to show that $\tau^*(\mathcal{F}) \leq \text{opt}/(k - \ell)$. For all $U \in \mathcal{F}^s$, any feasible solution H to Rooted Node-GSN has at least $k - \ell$ edges covering U , by Menger’s Theorem. Thus if x is a characteristic vector of $E(H)$, then $x/(k - \ell)$ is a feasible solution for the LP-relaxation for edge-covering \mathcal{F}^s . The statement follows. ■

Node costs: At iteration ℓ , the node-cost of the edge set I computed is: $O(\alpha \log |T|) \cdot \text{opt}$ in the case of Element-GSN, and $O(\alpha \ell \log |T|) \cdot \text{opt}$ in the case of Rooted GSN; here α is the parameter in Assumption 2. Hence after k iterations, the total node-costs is: $O(\alpha k \log |T|) \cdot \text{opt}$ in the case of Element-GSN, and $O(\alpha k^2 \log |T|) \cdot \text{opt}$ in the case of Rooted GSN, where $1 \leq \alpha \leq 2$.

As for GSN with node costs, it is remarked in [7] that a γ -approximation algorithm for Element-GSN implies an $O(k^3 \gamma \log |T|)$ -approximation algorithm for Node-GSN. Thus for node costs, our $O(k \log |T|)$ -approximation for Element-GSN together with the result of [7], implies an $O(k^4 \log^2 |T|)$ -approximation algorithm for Node-GSN. This finishes the proof of Theorem 1.4.

5. CONCLUSIONS

In this paper we developed approximation algorithm for GSN problems for both edge and node costs. Our

algorithms are simple, combinatorial, and achieve much better approximation guarantees than the previously known ones. For edge costs, our ratios are $O(k \ln k)$ for Rooted GSN with requirements in $\{0, k\}$, $O(k^2)$ for Rooted GSN with arbitrary requirements, and $O(k^2 \ln k)$ for Subset k -Connected Subgraph. These ratios are constants for bounded values of k . For node costs, we gave the first non-trivial algorithm for Element-GSN, with ratio matching the best known one for Edge-GSN. We believe that our ratios, except the ratio for general GSN with node costs, will not be easy to improve. However, for edge costs, it is likely that Rooted GSN with requirements in $\{0, k\}$ admits an $O(k)$ -approximation algorithm. Such algorithm would probably rely on the iterative rounding method, while all algorithms in this paper are combinatorial.

The notion of independence presented in this paper has a natural (but non-trivial) extension to GSN with arbitrary requirements. We believe that the method presented in this paper will lead eventually to ratio for GSN that depends on k only, most likely $O(k^3)$.

Finally, we note that for all problems considered, there are good evidences that they are unlikely to admit a polylogarithmic ratio for large values of k . Furthermore, a significant obstacle lies in the way of achieving a ratio sublinear in k . As was mentioned, by [18], this would imply a sublinear in n ratio for the *directed* variant. Such algorithms are known only for $k = 1$ and they are highly non-trivial; see [3] for the rooted case and [10] for the general case. Furthermore, for the directed variant, even for rooted requirements, no ratio better than the trivial $O(n)$ is known even for $k = 2$.

REFERENCES

- [1] A. Agrawal, P. N. Klein, and R. Ravi, "When trees collide: An approximation algorithm for the generalized Steiner problem on networks," *SIAM J. Comput.*, vol. 24, no. 3, pp. 440–456, 1995.
- [2] T. Chakraborty, J. Chuzhoy, and S. Khanna, "Network design for vertex connectivity," in *STOC*, 2008, pp. 167–176.
- [3] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner problems," *Journal of Algorithms*, vol. 33, pp. 73–91, 1999.
- [4] C. Chekuri and N. Korula, "Single-sink network design with vertex connectivity requirements," in *FSTTCS*, 2008.
- [5] J. Cheriyan, S. Vempala, and A. Vetta, "Network design via iterative rounding of setpair relaxations," *Combinatorica*, vol. 26, no. 3, pp. 255–275, 2006.
- [6] J. Cheriyan and A. Vetta, "Approximation algorithms for network design with metric costs," *SIAM J. Discr. Math.*, vol. 21, no. 3, pp. 612–636, 2007.
- [7] J. Chuzhoy and S. Khanna, "An $O(k^3 \log n)$ -approximation algorithms for vertex-connectivity network design," to appear in *FOCS* 2009.
- [8] —, "Algorithms for single-source vertex-connectivity," in *FOCS*, 2008, pp. 105–114.
- [9] J. Fackharoenphol and B. Laekhanukit, "An $O(\log^2 k)$ -approximation algorithm for the k -vertex connected subgraph problem," in *STOC*, 2008, pp. 153–158.
- [10] M. Feldman, G. Kortsarz, and Z. Nutov, "Improved approximating algorithms for directed steiner forest," in *SODA*, 2009, pp. 922–931.
- [11] L. K. Fleischer, K. Jain, and D. P. Williamson, "An iterative rounding 2-approximation algorithm for the element connectivity problem," in *FOCS*, 2001, pp. 339–347.
- [12] A. Frank and T. Jordán, "Minimal edge-coverings of pairs of sets," *J. on Comb. Theory B*, vol. 65, pp. 73–110, 1995.
- [13] M. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, E. Tardos, and D. Williamson, "Improved approximation algorithms for network design problems," in *SODA*, 1994, pp. 223–232.
- [14] K. Jain, "A factor 2 approximation algorithm for the generalized Steiner network problem," *Combinatorica*, vol. 21, no. 1, pp. 39–60, 2001.
- [15] P. N. Klein and R. Ravi, "A nearly best-possible approximation algorithm for node-weighted steiner trees," *Journal of Algorithms*, vol. 19, no. 1, pp. 104–115, 1995.
- [16] G. Kortsarz, R. Krauthgamer, and J. R. Lee, "Hardness of approximation for vertex-connectivity network design problems," *SIAM J. on Comput.*, vol. 33, no. 3, pp. 704–720, 2004.
- [17] G. Kortsarz and Z. Nutov, *Approximating minimum cost connectivity problems*, Ch. 58 in *Approximation Algorithms and Metaheuristics*, T. F. Gonzales ed., PWS, 2007.
- [18] Y. Lando and Z. Nutov, "Inapproximability of survivable networks," *Theoretical Computer Science*, vol. 410, no. 21–23, pp. 2122–2125, 2009.
- [19] Z. Nutov, "Approximating node-connectivity augmentation problems," to appear in *APPROX* 2009.
- [20] —, "A note on rooted survivable networks," to appear in *Information Processing Letters*.
- [21] —, "Approximating connectivity augmentation problems," in *SODA*, 2005, pp. 176–185, to appear in *Transactions on Algorithms*.
- [22] —, "Approximating steiner networks with node weights," in *LATIN*, 2008, pp. 411–422.
- [23] —, "An almost $O(\log k)$ -approximation for k -connected subgraphs," in *SODA*, 2009, pp. 912–921.
- [24] R. Ravi and D. P. Williamson, "An approximation algorithm for minimum-cost vertex-connectivity problems," *Algorithmica*, vol. 18, pp. 21–43, 1997.