A Note on Rooted Survivable Networks*

Zeev Nutov

Dept. of Computer Science

The Open University of Israel

nutov@openu.ac.il

Abstract

The (undirected) Rooted Survivable Network Design (Rooted SND) problem is: given a complete graph on node set V with edge-costs, a root $s \in V$, and (node-)connectivity requirements $\{r(t) : t \in T \subseteq V\}$, find a minimum cost subgraph G that contains $r(t)$ internally-disjoint st-paths for all $t \in T$. For large values of $k = \max_{t \in T} r(t)$ Rooted SND is at least as hard to approximate as Directed Steiner Tree [Lando & Nutov, APPROX 2008]. For Rooted SND [Chuzhoy & Khanna, FOCS 08] gave recently an approximation algorithm with ratio $O(k^2 \log n)$. Independently, and using different techniques, we obtained at the same time a simpler primal-dual algorithm with the same ratio.

1 Introduction

Let $\kappa_G(u, v)$ denote the maximum number of internally-disjoint uv-paths in a graph G. We consider the rooted variant of the following fundamental problem in network design:

Survivable Network Design (SND):

Instance: A complete graph on node-set V with edge-costs $c(e)$ and connectivity requirements $\{r(u, v) : u, v \in V\}$.

Objective: Find a minimum cost subgraph $G = (V, E)$ so that $\kappa_G(u, v) \geq r(u, v)$ for all $u, v \in V$.

If all the requirement are "rooted", namely from a specific node s, then we have the following important particular case of SND:

Rooted SND:

Instance: A complete graph on node set V with edge-costs $c(e)$, a root $s \in V$, a set T of *terminals*, and requirements $\{r(t) > 0 : t \in T\}$.

Objective: Find a minimum cost subgraph $G = (V, E)$ so that $\kappa_G(s, t) \geq r(t)$ for all $t \in T$.

*Preliminary version of this paper is a part of [20].

All graphs are assumed to be undirected and simple, unless stated otherwise. For an instance of SND at hand, let opt denote the optimal solution value, let $k = \max_{u,v \in V} r(u,v)$ denote the maximum requirement, and let $n = |V|$.

While the edge-connectivity case – the so called Steiner Network problem – admits a 2-approximation algorithm [11], up to recently non-trivial approximation algorithms for SND were known only for metric costs [6] by Cheriyan and Vetta, and for $0,1$-costs [14, 18, 17]. A hardness result of Kortsarz et al. [12] suggests that for general costs SND is unlikely to admit a polylogarithmic approximation; this is so even when the input graph is complete and the costs are in $0,1$ [19]. 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. Y. Lando and the author [15] proved that for $k = n/2 + k'$ SND is harder to approximate than its directed variant with maximum requirement k'. This is so also for Rooted SND, thus Rooted SND with $k > n/2$ is at least as hard to approximate as the notorious Directed Steiner Tree problem; a long standing best known ratio for the latter is $O(|T|^\varepsilon/\varepsilon^3)$ in $O(|T|^{4/\varepsilon} n^{2/\varepsilon})$ time [3]. For $k' = 2$ no sublinear approximation for the directed rooted variant is known.

Some variants of SND were extensively studied, and in particular the k-Connected Subgraph problem, which is the variant of SND with $r(u,v) = k$ for all $u,v \in V$; see [20] for the best known ratio for this problem and the references therein. We refer the reader to [13] for a survey on various connectivity problems, and here mention some literature relevant to this paper. Recently, Rooted SND received some attention, because Chakraborty, Chuzhoy, and Khanna [2] obtained a $k^{O(k^2)} \log^4 n$-approximation for it; prior to this, there was almost no literature on Rooted SND, except for the case of $0,1$-costs [14, 18, 17]. Slightly later the ratio was improved to $k^{O(k)} \log n$ by Chekuri & Korula [5], and then to $O(k^2 \log n)$ by Chuzhoy and Khanna [7]. Independently, we obtained at the same time a much simpler primal-dual algorithm with the same ratio $O(k^2 \log n)$.

Theorem 1.1 Rooted SND *admits an* $O(k^2 \log |T|)$*-approximation algorithm,* $k = \max_{t \in T} r(t)$.

We note that slightly later two additional different $O(k^2 \log n)$-approximation algorithms were suggested by Chuzhoy and Kanna [8] and Chekuri and Korula [4]. The algorithm in [8] relies on the iterative rounding algorithm of [9] for the so called element-connectivity problem, and is not combinatorial. The algorithm of [4] relies on the non-trivial machinery developed in [4]. Our algorithm is combinatorial, and our proof of Theorem 1.1 is relatively simple while being self contained. Moreover, the algorithm presented in this paper was recently generalized by the author in [16] to achieve the currently best knowm ratio $O(k^2)$ for Rooted SND. We also note that in [8], Chuzhoy and Khanna gave an $O(k^3 \log n)$-approximation algorithm for SND with arbitrary requirements, which is the currently best known ratio for the problem.

As an intermediate problem, we consider the Rooted SND Augmentation problem, which is the restriction of Rooted SND to instances in which G contains a subgraph $J = (V, E_J)$ of cost 0 so that $\kappa_J(t,s) = \ell$ and $r(t) = \ell + 1$ for all $t \in T$; namely, we seek to increase at minimum cost the connectivity between s and the nodes in T from $k - 1 = \ell$ to $k = \ell + 1$.

Theorem 1.2 Rooted SND Augmentation *admits an $O(\ell \log |T|)$-approximation algorithm.*

It is easy to see that if Rooted SND Augmentation admits a ρ-approximation algorithm then Rooted SND admits a $k\rho$-approximation algorithm; thus Theorem 1.1 follows from Theorem 1.2, so we only need to prove Theorem 1.2. To see this, consider the following algorithm for Rooted SND. Start with $J = (V, \emptyset)$ and continue with iterations. Iteration ℓ starts with a graph J with $\kappa_J(s,t) = \min\{\ell, r(t)\}$ for all $t \in T$, and seeks to increase the st-connectivity from ℓ to $\ell+1$ for every $t \in T$ with $\kappa_J(s,t) = \ell$ and $r(t) \geq \ell + 1$; thus this is an instance of Rooted SND Augmentation. We find an edge set I_ℓ of cost $\rho \cdot \mathsf{opt}$ using the ρ-approximation algorithm for Rooted SND Augmentation. After at most k iterations J satisfies the requirements, and its cost is $\leq k\rho \cdot \mathsf{opt}$.

Remark: For Rooted SND with requirements in $\{0, k\}$, namely, when $r(t) = k$ for all $t \in T$, our algorithm has ratio $O(k \log k \cdot \log |T|)$. This can be proved using standard LP scaling techniques, by showing that at iteration ℓ the cost of the subgraph computed is $O(\mathsf{opt}/(k - \ell))$. For this restricted version, Chuzhoy and Khanna [7] obtained the slightly better ratio of $O(k \log |T|)$.

2 Proof of Theorem 1.2

Let $\Gamma_J(X) = \Gamma(X) = \{v \in V - X : uv \in E_J \text{ for some } u \in X\}$ denote the set of *neighbors* of X in J, and let $X^* = V - (X \cup \Gamma_J(X))$. To avoid considering "mixed" cuts that contain both nodes and edges, we may assume that $st \notin E_J$ for all $t \in T$. One way to achieve this is to subdivide every edge $st \in E_J$ with $t \in T$ by a new node.

Definition 2.1 *A node subset $X \subseteq V$ is* t-tight *for $t \in T$ if $t \in X$, $s \in X^*$, and $|\Gamma_J(X)| = \ell$; X is* tight *if it is t-tight for some $t \in T$. A tight set is a* core *if it does not contain two inclusion minimal tight sets. An inclusion-minimal (inclusion-maximal) core is a* min-core *(*max-core*). Let $\mathcal{C}_J = \{C_1, \ldots, C_\nu\}$ denote the set of min-cores in J.*

We say that an edge e covers a tight set X if it has one endnode in X and the other in X^*. By Menger's Theorem, I is a feasible solution to an instance of Rooted SND Augmentation if, and only if, I covers all tight sets (assuming $st \notin E_J$ for all $t \in T$). Thus our goal is to find such I of low cost. Note that terminals not belonging to any min-core can be discarded, as any tight set containing such a terminal also contains a terminal that belongs to some min-core. Hence from now and on we assume that every terminal belongs to some min-core.

The following "sub-modular" and "posi-modular" properties of the function $\Gamma(\cdot) = \Gamma_J(\cdot)$ are well known, c.f., [19].

Proposition 2.1 *For any $X, Y \subseteq V$ the following holds:*

$$|\Gamma(X)| + |\Gamma(Y)| \geq |\Gamma(X \cap Y)| + |\Gamma(X \cup Y)| \tag{1}$$

$$|\Gamma(X)| + |\Gamma(Y)| \geq |\Gamma(X \cap Y^*)| + |\Gamma(Y \cap X^*)| \tag{2}$$

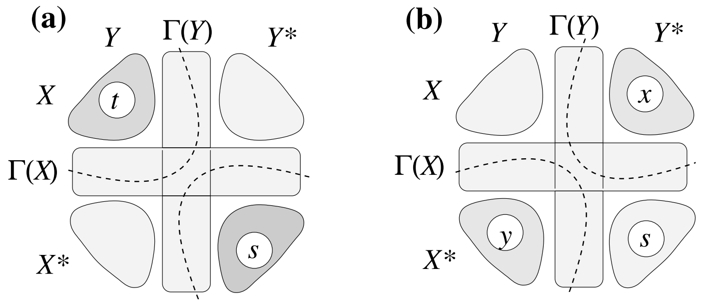

Figure 1: Illustration to the proof of Lemma 2.2.

$$(X \cap Y)^* = X^* \cup Y^*, \ (X \cup Y)^* = X^* \cap Y^* \qquad \textit{if equality holds in } (1) \tag{3}$$

$$(X \cap Y^*)^* = X^* \cup Y, \ (Y \cap X^*)^* = X \cup Y^* \qquad \textit{if equality holds in } (2) \tag{4}$$

Lemma 2.2 *Let X be x-tight and let Y be y-tight. Then:*

(i) *If $x \in X \cap Y$ then $X \cap Y, X \cup Y$ are x-tight, and if $y \in X \cap Y$ then $X \cap Y, X \cup Y$ are y-tight. Furthermore, in both cases equality holds in* (1)*.*

(ii) *If $x \in X \cap Y^*, y \in Y \cap X^*$ then $X \cap Y^*$ is x-tight, $Y \cap X^*$ is y-tight, and equality holds in* (2)*.*

(iii) *If none of* (i),(ii) *holds then $y \in \Gamma(X)$ or $x \in \Gamma(Y)$.*

Proof: Part (iii) is obvious, so we prove parts (i),(ii) using (1) and (2), see Figure 2.

If $x \in X \cap Y$ (the proof of the case $y \in X \cap Y$ is similar) then by (1) (see Figure 2(a)):

$$\ell + \ell = |\Gamma(X)| + |\Gamma(Y)| \geq |\Gamma(X \cap Y)| + |\Gamma(X \cup Y)| \geq \ell + \ell \ .$$

Hence equality holds everywhere, so $X \cap Y, X \cup Y$ are x-tight.

If $x \in X \cap Y^*$ and $y \in Y \cap X^*$ then by (2) (see Figure 2(b)):

$$\ell + \ell = |\Gamma(X)| + |\Gamma(Y)| \geq |\Gamma(X \cap Y^*)| + |\Gamma(Y \cap X^*)| \geq \ell + \ell \ .$$

Hence equality holds everywhere, so $X \cap Y^*$ is x-tight and $Y \cap X^*$ is y-tight. $\square$

In [10], a set-family $\mathcal{F}$ was called *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 e was said to cover a set X if e has one endnode in X and the other in $V - X$. In [10] is given a 2-approximation primal-dual algorithm that computes a cover of an uncrossable family $\mathcal{F}$. In our case, we use the following modified definition:

Definition 2.2 *A subfamily $\mathcal{F}$ of tight sets is* bi-uncrossable *if for any $X, Y \in \mathcal{F}$ at least one of the following holds: $X \cap Y, X \cup Y \in \mathcal{F}$ and* (3) *holds, or $X \cap Y^*, Y \cap X^* \in \mathcal{F}$ and* (4) *holds.*

As we will show in the next section (to obtain a combinatorial and fast implementation), the primal-dual algorithm of [10] can be adjusted to cover "setpair" families $\{\{X, X^*\} : X \in \mathcal{F}\}$, provided $\mathcal{F}$ is bi-uncrossable. Alternatively, in [9] a 2-approximation algorithm was given for a much more general setpair cover problem, generalizing the iterative rounding method of Jain [11].

Unfortunately, the family of tight sets may not be bi-uncrossable, but we will show a method to decompose it into bi-uncrossable families.

Lemma 2.3 *For any tight set X and any $C_i \in \mathcal{C}_J$ either $C_i \cap X \cap T = \emptyset$ or $(C_i \cap T) \subseteq X$. Thus $C_i \cap C_j \cap T = \emptyset$ for any $i \neq j$.*

Proof: Otherwise, $C_i \cap X$ is tight, by Lemma 2.2 (i), contradicting the minimality of C_i. $\square$

Definition 2.3 *For $i = 1, \ldots \nu$, let $T_i = T \cap C_i$, let M_i be some max-core containing C_i, and let $\Gamma_i = \Gamma(M_i)$. Let $\mathcal{M}_J = \{M_1, \ldots, M_\nu\}$. We say that $M_i, M_j \in \mathcal{M}_J$ are* independent *if the sets $T_i \cap M_j^*, T_j \cap M_i^*$ are both nonempty.*

From Lemmas 2.2 and 2.3 we have:

Corollary 2.4 *For any i the set M_i is unique. For any $i \neq j$, at least one of the following holds: M_i, M_j are independent and thus $M_i \cap M_j^*, M_j \cap M_i^*$ are tight, or $T_i \subseteq \Gamma_j$, or $T_j \subseteq \Gamma_i$.*

We note that the families $\mathcal{C}_J$ and $\mathcal{M}_J$ can be computed using $O(|T|)$ max-flow computations as follows. It is well known that given $t \in T$, one max-flow computation suffices to find the unique minimal t-tight set C_t, and the unique maximal t-tight set, or to determine that such sets do not exist, e.g. see [18]. The family $\mathcal{C}_J$ is formed by the inclusion minimal members of the family $\{C_t : t \in T\}$. To find the max-core M_i that contains a specific min-core C_i do the following. Add to J an edge from s to every min-core C_j distinct from C_i. The added edges do not cover any core containing C_i, but they cover all the other tight sets. Thus in the obtained graph, M_i is the largest t-tight set for any $t \in T_i$.

Given a subfamily $\mathcal{M} \subseteq \mathcal{M}_J$, the subfamily of tight sets induced by $\mathcal{M}$ is

$$\mathcal{F}(\mathcal{M}) = \{X : X \subseteq M \in \mathcal{M}, X \text{ is tight}\} .$$

Lemma 2.5 *If $X, Y \subseteq M_i$ are tight then $X \cap Y, X \cup Y$ are also tight, and* (3) *holds for X, Y. If M_i, M_j with $i \neq j$ are independent then for any tight $X \subseteq M_i$ and $Y \subseteq M_j$ the sets $X \cap Y^*, Y \cap X^*$ are tight and* (4) *holds for X, Y. Thus if the members of $\mathcal{M}$ are pairwise independent, then the family $\mathcal{F}(\mathcal{M})$ is uncrossable.*

Proof: The first statement follows from Lemma 2.2 (i). The second statement follows from Lemma 2.2 (ii) and the fact that if $T_i \cap M_j^* \neq \emptyset$ then $T_i \cap Y^* \neq \emptyset$ for any $Y \subseteq M_j$. $\square$

Lemma 2.6 *The family $\mathcal{M}_J$ can be partitioned into at most $2\ell + 1$ parts so that the members of each part are pairwise independent, and such a partition can be found in polynomial time.*

Proof: Construct an auxiliary directed graph $\mathcal{J}$ as follows. The node set of $\mathcal{J}$ is $\mathcal{M}_J$. Add an arc M_iM_j if $T_i \subseteq \Gamma_j$. The maximum indegree of every node in $\mathcal{J}$ is $\leq \ell$. This implies that every subgraph of the underlying graph of $\mathcal{J}$ has a node of degree $\leq 2\ell$. A graph is d-degenerate if every subgraph of it has a node of degree at most d. It is well known that any d-degenerate graph is $(d+1)$-colorable, and such coloring can be computed in polynomial time. Hence $\mathcal{J}$ is $(2\ell+1)$-colorable, thus its node set can be partitioned into at most $2\ell+1$ independent sets. $\square$

Lemma 2.7 *If I covers $\mathcal{F}(\mathcal{M}_J)$ then the number of min-cores in $G+I$ is at most $\nu/2$.*

Proof: Every min-core of $J+I$ is a tight set in J. Thus by Lemma 2.3 and by the definition of M_i every min-core C of $J+I$ contains the terminals of at least 2 distinct min-cores C_i, C_j of J; namely, $T_i, T_j \subseteq C$. As the min-cores of $J+I$ are also disjoint on the terminals, by Lemma 2.3, the statement follows. $\square$

Summarizing, we can find an edge set I of cost $\leq 2(2\ell+1)\cdot \mathsf{opt}$ so that the number of min-cores in $G+I$ is $\leq \nu/2$. Such I is a union of 2-approximate covers of $2\ell+1$ uncrossable families as in Lemma 2.6. We can apply this procedure iteratively, until no min-cores remain. The number of iterations is at most $1+\log\nu = O(\log|T|)$, and the overall cost over all iterations is $O(\ell \log |T|)\cdot\mathsf{opt}$.

The proof of Theorem 1.2, and thus also the proof of Theorem 1.1 is now complete.

3 Algorithm for covering bi-uncrossable families

We start with recalling some definitions. Let $\mathcal{F}$ be a set-family on a groundset V, so that $\emptyset, V \notin \mathcal{F}$. Suppose that for every $S \in \mathcal{F}$ corresponds a unique nonempty set $S^* \subseteq V - S$. We assume that $\mathcal{F}$ is bi-uncrossable (w.r.t. the mapping $S \to S^*$), see Definition 2.2. Let E be an edge set on V with costs $\{c(e) : e \in E\}$. For $I \subseteq E$ and $S \in \mathcal{F}$ let $\delta_I(S)$ denote the set of edges in I with one endnode in S and the other in S^*, and let $\delta(S) = \delta_E(S)$. An edge e *covers* a set $S \in \mathcal{F}$ if it has one endnode in S and the other endnode in S^*.

Lemma 3.1 *Let $\mathcal{C}(\mathcal{F})$ denote the family of minimal members in a bi-uncrossable family $\mathcal{F}$. Then:*
(i) *$X \subseteq Y$ implies $Y^* \subseteq X^*$ for any $X, Y \in \mathcal{F}$.*
(ii) *If $C \in \mathcal{C}(\mathcal{F})$ and $X \in \mathcal{F}$, then $C \subseteq X$ or $C \subseteq X^*$; thus the sets in $\mathcal{C}(\mathcal{F})$ are pairwise disjoint.*

Proof: For Part (i), note that if $X \subseteq Y$ then $X \cap Y^* = \emptyset \notin \mathcal{F}$, hence we must have $X \cap Y, X \cup Y \in \mathcal{F}$ and (3) holds. Consequently, by (3), $Y^* = (X \cup Y)^* = X^* \cap Y^* \subseteq X^* \cup Y^* = (X \cap Y)^* = X^*$. We prove Part (ii). If (3) holds for C, X, then $C \cap X \in \mathcal{F}$. If (4) holds for C, X, then $C \cap X^* \in \mathcal{F}$. By the minimality of C, we must have $C \subseteq X$ in the former case and $C \subseteq X^*$ in the latter case. $\square$

We describe a combinatorial 2-approximation algorithm for the problem of finding a minimum cost edge-cover of a bi-uncrossable family $\mathcal{F}$. The algorithm and the proof of the approximation ratio are along the lines of (a simplification of) the ones given in [10] for the case $X^* = V - X$, using the primal-dual method. Alternatively, any LP interpretation can be avoided using the local ratio technique of Bar-Yehuda and Rawitz [1].

Consider the following LP-relaxation for our problem **(P)** and its dual program **(D)**:

$$
\textbf{(P)}\quad \begin{array}{lll} \min & \sum_{e\in E} c_e x_e & \\ \text{s.t.} & \sum_{e\in\delta(S)} x_e \geq 1 & \forall S \in \mathcal{F} \\ & x_e \geq 0 & \forall e \in E \end{array}
\qquad\qquad
\textbf{(D)}\quad \begin{array}{lll} \max & \sum_{S\in\mathcal{F}} y_S & \\ \text{s.t.} & \sum_{\delta(S)\ni e} y_S \leq c_e & \forall e \in E \\ & y_S \geq 0 & \forall S \in \mathcal{F}. \end{array}
$$

We now describe the algorithm. Given a solution y to **(D)**, an edge $e \in E$ is *tight* if the corresponding inequality in **(D)** holds with equality. The algorithm has two phases. During the algorithm, for a partial solution I, let $\mathcal{F}_I$ denote the minimal members of $\mathcal{F}$ not covered by I. It is easy to see that if $\mathcal{F}$ is uncrossable, so is $\mathcal{F}_I$, for any I.

Phase 1 starts with $I = \emptyset$ an applies a sequence of *iterations*. At the beginning of an iteration, we compute the family $\mathcal{C}(\mathcal{F}_I)$. Then we raise the dual variables corresponding to the members of $\mathcal{C}(\mathcal{F}_I)$ uniformly until some edge $e \in E - I$ becomes tight, and add e to I. Phase I terminates when $\mathcal{C}(\mathcal{F}_I) = \emptyset$, namely when all $\mathcal{F}$ is covered.

Phase 2 applies on I "reverse delete", which means the following. Let $I = \{e_1, \ldots, e_j\}$, where e_{i+1} was added after e_i. For $i = j$ downto 1, we delete e_i from I if $I - e_i$ still covers $\mathcal{F}$. At the end of the algorithm, I is output.

It is easy to see that the produced dual solution is feasible, hence $\sum_{S\in\mathcal{F}} y_S \leq \mathsf{opt}$, by the Weak Duality Theorem. We prove that at the end of the algorithm $\sum_{e\in I} c(e) \leq 2\sum_{S\in\mathcal{F}} y_S$. As any edge in I is tight, this is equivalent to $\sum_{e\in I}\sum_{\delta(S)\ni e} y_S \leq 2\sum_{S\in\mathcal{F}} y_S$. By changing the order of summation we get:

$$\sum_{S\in\mathcal{F}} |\delta_I(S)| y_S \leq 2\sum_{S\in\mathcal{F}} y_S \ .$$

It is sufficient to prove that at any iteration the increase at the left hand side is at most the increase in the right hand side. Let us fix some iteration, and let $\mathcal{C}$ be the the family of minimal sets among the members of $\mathcal{F}$ not yet covered. The increase in the left hand side is $\varepsilon \cdot \sum_{C\in\mathcal{C}} |\delta_I(C)|$, where ε is the amount by which the dual variables were raised in the iteration, while the increase in the right hand side is $\varepsilon \cdot 2|\mathcal{C}|$. Consequently, it is sufficient to prove that $\sum_{C\in\mathcal{C}} |\delta_I(C)| \leq 2|\mathcal{C}|$. As the edges were deleted in reverse order, the set I' of edges in I that were added after the iteration (and "survived" the reverse delete phase), form an inclusion minimal edge-cover of the family $\mathcal{F}'$ of members in $\mathcal{F}$ that are uncovered at the beginning of the iteration. Note also that $\bigcup_{C\in\mathcal{C}} \delta_I(C) \subseteq I'$. Hence to prove the ratio of 2, it is sufficient to prove the following purely combinatorial statement, in which we revise our notation to $\mathcal{F} \leftarrow \mathcal{F}'$ and $I \leftarrow I'$.

Lemma 3.2 *Let I be an inclusion minimal edge-cover of an uncrossable family $\mathcal{F}$ and let $\mathcal{C} = \mathcal{C}(\mathcal{F})$*

be the family of inclusion minimal members of $\mathcal{F}$. Then

$$\sum_{C \in \mathcal{C}} |\delta_I(C)| \leq 2|\mathcal{C}| - 1 \ .$$

We note that the sufficiency of Lemma 3.2 for proving a ratio of 2 also follows from a local ratio argument of Bar-Yehuda and Rawitz [1]. In the rest of this section we prove Lemma 3.2.

Definition 3.1 *We say that an edge set F is a* fit-cover *of a subfamily $\mathcal{L} \subseteq \mathcal{F}$, or that $\mathcal{L}$ is a* fit-family *for F, if $|\mathcal{L}| = |F|$ and for every $e \in F$ there is a* fit-set *$S_e \in \mathcal{L}$ so that $\delta_F(S_e) = \{e\}$; namely, e is the unique edge in F that covers S_e.*

A set-family $\mathcal{L} \subseteq \mathcal{F}$ is *bi-laminar* (w.r.t. the mapping $S \to S^*$) 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 $X \cap Y = \emptyset$.

Lemma 3.3 *Let I be an inclusion minimal cover of a bi-uncrossable family $\mathcal{F}$. Then there exists a bi-laminar family $\mathcal{L} \subseteq \mathcal{F}$ so that I is a fit-cover of $\mathcal{L}$.*

Proof: By the minimality of I, for every $e \in I$ there exists $S_e \in \mathcal{F}$ such that e is the unique edge in I that covers S_e. Thus there exists $\mathcal{L} \subseteq \mathcal{F}$ so that I is a fit-cover of $\mathcal{L}$. We prove that there exists such bi-laminar $\mathcal{L}$. Among all fit-families for I contained in $\mathcal{F}$, let $\mathcal{L}$ be one with $\sum_{S \in \mathcal{L}} |S|$ minimal. We claim that $\mathcal{L}$ is bi-laminar. Let $X, Y \in \mathcal{L}$, where X is a fit-set for e and Y is a fit-set for f. We claim that then at least one of the following holds:

(i) If $X \cap Y, X \cup Y \in \mathcal{F}$ and (3) holds, then $X \cap Y$ is a fit-set for one of e, f (and $X \cup Y$ is a fit-set for the other); thus $\mathcal{L} - \{X\} + \{X \cap Y\}$ or $\mathcal{L} - \{Y\} + \{X \cap Y\}$ is also a fit-family for I. Consequently, we must have $X \subset Y$ or $Y \subset X$ in this case, by the choice of $\mathcal{L}$.

(ii) If $X \cap Y^*, Y \cap X^* \in \mathcal{F}$ and (4) holds, then $X \cap Y^*$ is a fit-set for one of e, f and $Y \cap X^*$ is a fit-set for the other; thus $\mathcal{L} - \{X, Y\} + \{X \cap Y^*, Y \cap X^*\}$ is also a fit-family for I. Consequently, we must have $X \subseteq Y^*$ and $Y \subseteq X^*$ in this case, by the choice of $\mathcal{L}$.

We prove that (i) or (ii) must hold. Suppose that $X \cap Y, X \cup Y \in \mathcal{F}$ and (3) holds; the proof when $X \cap Y^*, Y \cap X^* \in \mathcal{F}$ and (4) holds is similar. Then there is an edge in I covering $X \cap Y$ and there is an edge in I covering $X \cup Y$. However, if (3) holds for X, Y, then if 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 $X \cap Y$ or $X \cup Y$, so one of e, f covers $X \cap Y$, and thus the other must cover $X \cup Y$. □

Let $F = \bigcup_{C \in \mathcal{C}} \delta_I(C)$. Note that $\delta_I(C) = \delta_F(C)$ for any $C \in \mathcal{C}$. As any subfamily of a bi-laminar family is also bi-laminar, we conclude from Lemma 3.3 that there exists a bi-laminar family $\mathcal{L} \subseteq \mathcal{F}$ so that F is a fit-cover of $\mathcal{L}$. Note that by Lemma 3.1 (ii), $\mathcal{L} \cup \mathcal{C}$ is bi-laminar, and for every $S \in \mathcal{L}$ there is $C \in \mathcal{C}$ so that $C \subseteq S$. Thus to finish the proof, it is sufficient to prove:

Lemma 3.4 *Let $\mathcal{L}, \mathcal{C}$ be set families, so that $\mathcal{L} \cup \mathcal{C}$ is bi-laminar, the members of $\mathcal{C}$ are pairwise disjoint, and for every $S \in \mathcal{L}$ there is $C \in \mathcal{C}$ so that $C \subseteq S$. Let F be an edge set that covers $\mathcal{C}$, fit-covers $\mathcal{L}$, and and so that every edge in F covers some $C \in \mathcal{C}$. Then*

$$\sum_{C \in \mathcal{C}} |\delta_F(C)| \leq 2|\mathcal{C}| - 1 \ . \tag{5}$$

We now prove Lemma 3.4. Let us say that $C \in \mathcal{C}$ is a *leaf-set* if $|\delta_F(C)| = 1$. The following statement shows, among others, that at least one leaf-set exists.

Claim 3.5 *Let S be an inclusion minimal member of $\mathcal{L}$, let $C \in \mathcal{C}$ so that $C \subseteq S$, and let $e \in F$ be an edge that covers C. Then $S = S_e$. In particular, C is a leaf-set.*

Proof: We cannot have $S_e \subset S$ by the minimality of S. We cannot have $S \subset S_e$, or $S \subseteq S_e^*$ and $S_e \subseteq S^*$, since then e covers both S, S_e, by Lemma 3.1 (i). The statement follows. $\square$

For $C \in \mathcal{C}$ let $\mathcal{L}_C$ be the family of sets in $\mathcal{L}$ that contain C and do not contain any other member of $\mathcal{C}$; note that $C \in \mathcal{L}_C$ if $C \in \mathcal{L}$. Let F_C be the set of the fit-edges of the members of $\mathcal{L}_C$.

Claim 3.6 $|\mathcal{L}_C| \leq 2$ *for any leaf-set C.*

Proof: Suppose to the contrary that there are sets $C \subseteq S_e \subset S_f \subset S_g$ in $\mathcal{L}_C$, where S_e is the inclusion minimal set in $\mathcal{L}$ with $C \subseteq S_e$. By Claim 3.5, e covers C. Now consider a set $C_f \in \mathcal{C}$ covered by f. We must have $C_f \neq C$, since otherwise f covers S_e, by Lemma 3.1 (i). By Lemma 3.1 (ii), $C_f \subseteq S_g$ or $C_f \subseteq S_g^*$. Since $S_g \in \mathcal{L}_C$, we must have $C_f \subseteq S_g^*$. However then f covers S_g, by Lemma 3.1 (i). This contradicts that g is the unique edge in F covering S_g. $\square$

Let C be a leaf-set as in Claim 3.5. A natural approach to prove Lemma 3.4 is by induction. Obtain a triple $\mathcal{C}', \mathcal{L}', F'$ by removing C from $\mathcal{C}$, $\mathcal{L}_C$ from $\mathcal{L}$, and F_C from F. Then in inequality (5), the decrease in the l.h.s. is at most 2, and the decrease in the r.h.s. is exactly 2. This implies the bound $\sum_{C \in \mathcal{C}} |\delta_F(C)| \leq 2|\mathcal{C}|$. The improved bound in Lemma 3.2 follows from the observation that if $\mathcal{C} = \{C\}$ then $|\delta_F(C)| = 1$, by Claim 3.5. This approach works, except one case: we need the triple $\mathcal{C}', \mathcal{L}', F'$ to satisfy the assumptions of Lemma 3.4. A problem can occur only when an edge $f \in F_C$ covers a leaf-set C' distinct from C, as then F' does not cover $C' \in \mathcal{C}$. In this case, we do not delete f, and add C' to $\mathcal{L}$ to be the fit-set for f instead of S_f.

Formally, the triple $\mathcal{C}', \mathcal{L}', F'$ is defined as follows. We set $\mathcal{C}' = \mathcal{C} - \{C\}$. Let S_e be the minimal and S_f the maximal set in $\mathcal{L}(C)$. Note that e covers C, and that $e = f$ and $S_e = S_f$ if $|\mathcal{L}_C| = 1$. If f covers a leaf set C' distinct from C, then we set $F' = (F - \{e\}) + \{f\}$ and $\mathcal{L}' = (\mathcal{L} - \{S_e\}) + \{C'\}$; otherwise, we set $F' = F - \{e, f\}$ and $\mathcal{L}' = \mathcal{L} - \{S_e, S_f\}$. It is immediate to see that $\mathcal{C}', \mathcal{L}', F'$ satisfy the assumptions of Lemma 3.4, and that $\sum_{C \in \mathcal{C}} |\delta_F(C)| - \sum_{C \in \mathcal{C}'} |\delta_{F'}(C)| \leq 2$ (note that an equality may hold, if f covers a member of $\mathcal{C} - \{C\}$ that is not a leaf-set). Together with the observation that both sides of (5) equal 1 if $|\mathcal{C}| = 1$, this implies the inequality (5).

This finishes the proof of Lemma 3.4, and thus the proof that the algorithm described in this section has ratio 2 is also complete.

Acknowledgment: I thank two anonymous referees for many useful comments.

References

[1] R. Bar-Yehuda and D. Rawitz. One for the price of two: a unified approach for approximating covering problems. *Algorithmica*, 37(2):131–144, 2000.

[2] T. Chakraborty, J. Chuzhoy, and S. Khanna. Network design for vertex connectivity. In *STOC*, pages 167–176, 2008.

[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*, 33:73–91, 1999.

[4] C. Chekuri and N. Korula. A graph reduction step preserving element-connectivity and applications. To appear in *ICALP* 2009.

[5] C. Chekuri and N. Korula. Single-sink network design with vertex connectivity requirements. In *FSTTCS*, 2008.

[6] J. Cheriyan and A. Vetta. Approximation algorithms for network design with metric costs. *SIAM J. Discrete Math.*, 21(3):612–636, 2007.

[7] J. Chuzhoy and S. Khanna. Algorithms for single-source vertex-connectivity. In *FOCS*, pages 105–114, 2008.

[8] J. Chuzhoy and S. Khanna. An $O(k^3 \log n)$-approximation algorithms for vertex-connectivity network design. Manuscript, 2009.

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

[10] M. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, E. Tardos, and D. Williamson. Improved approximation algorithms for network design problems. In *SODA*, pages 223–232, 1994.

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

[12] G. Kortsarz, R. Krauthgamer, and J. R. Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM Journal on Computing*, 33(3):704–720, 2004.

[13] G. Kortsarz and Z. Nutov. Approximating minimum-cost connectivity problems. In T. F. Gonzalez, editor, Chapter 58 in *Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC, 2007.

[14] G. Kortsarz and Z. Nutov. Tight approximation algorithm for connectivity augmentation problems. *J. Comput. Syst. Sci*, 74(5):662–670, 2008.

[15] Y. Lando and Z. Nutov. Inapproximability of survivable networks. *Theor. Comput. Sci.*, 410(21-23):2122–2125, 2009.

[16] Z. Nutov. Approximating minimum cost connectivity problems via uncrossable bifamilies and spider-cover decompositions. To appear in *FOCS* 2009.

[17] Z. Nutov. Approximating node-connectivity augmentation problems. To appear in *APPROX* 2009.

[18] Z. Nutov. Approximating rooted connectivity augmentation problems. *Algorithmica*, 44:213–231, 2003.

[19] Z. Nutov. Approximating connectivity augmentation problems. In *SODA*, pages 176–185, 2005. To appear in *Transactions on Algorithms*.

[20] Z. Nutov. An almost $O(\log k)$-approximation for k-connected subgraphs. In *SODA*, pages 912–921, 2009.