Theoretical Computer Science 411 (2010) 2502–2512

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Approximating minimum power covers of intersecting families and directed edge-connectivity problems[☆]

Zeev Nutov [*]

The Open University of Israel, Raanana, Israel

ARTICLE INFO

Article history:
Received 22 July 2006
Received in revised form 13 February 2010
Accepted 2 March 2010
Communicated by D. Peleg

Keywords:
Approximation algorithms
Wireless networks
Power minimization
Directed graphs
Edge-connectivity
Intersecting families

ABSTRACT

Given a (directed) graph with costs on the edges, the power of a node is the maximum cost of an edge leaving it, and the power of the graph is the sum of the powers of its nodes. Let $\mathcal{G} = (V, \mathcal{E})$ be a graph with edge costs $\{c(e) : e \in \mathcal{E}\}$ and let k be an integer. We consider problems that seek to find a min-power spanning subgraph G of $\mathcal{G}$ that satisfies a prescribed edge-connectivity property. In the Min-Power k-Edge-Outconnected Subgraph problem we are given a root $r \in V$, and require that G contains k pairwise edge-disjoint rv-paths for all $v \in V - r$. In the Min-Power k-Edge-Connected Subgraph problem G is required to be k-edge-connected. For $k = 1$, these problems are at least as hard as the Set-Cover problem and thus have an $\Omega(\ln |V|)$ approximation threshold. For $k = \Omega(n^{\varepsilon})$, they are unlikely to admit a polylogarithmic approximation ratio [15]. We give approximation algorithms with ratio $O(k \ln |V|)$. Our algorithms are based on a more general $O(\ln |V|)$-approximation algorithm for the problem of finding a min-power directed edge-cover of an intersecting set-family; a set-family $\mathcal{F}$ is intersecting if $X \cap Y, X \cup Y \in \mathcal{F}$ for any intersecting $X, Y \in \mathcal{F}$, and an edge set I covers $\mathcal{F}$ if for every $X \in \mathcal{F}$ there is an edge in I entering X.

1. Introduction and preliminaries

A large research effort focused on developing algorithms for finding a "cheap" sub-network (subgraph) that satisfies prescribed requirements. In wired networks, where connecting any two nodes incurs a cost, the goal is to find a subgraph of the minimum cost. In wireless networks, a range (power) of the transmitters determines the resulting communication network. We consider finding a power assignment to the nodes of a network such that the resulting communication network satisfies prescribed edge-connectivity properties and the total power is minimized. We note that node-connectivity is more central here than edge-connectivity, as it models station crashes. For further motivation and applications to wireless networks see, e.g., [1,2,9,13,3,4,10,17].

Henceforth, unless stated otherwise, "graph" means "directed graph". Let $G = (V, E)$ be a (directed) graph with edge costs $\{c(e) : e \in E\}$. For $v \in V$, the *power* $p(v) = p_c(v)$ *of* v in G (w.r.t. c) is the maximum cost of an edge leaving v in G (or zero, if no such edge exists). The power $p(G) = \sum_{v \in V} p(v)$ of G is the sum of the powers of its nodes. Note that the ratio between the power $p(G)$ and the cost $c(G)$ of G can be as large as the maximum outdegree of a node in G, e.g., for stars with unit costs. The following known statement that appeared in various papers, c.f., [9,10], shows that this is the extremal case for general edge costs.

Proposition 1.1. *For any directed graph G holds: $c(G)/\Delta(G) \le p(G) \le c(G)$, where $\Delta(G)$ is the maximum outdegree of a node in G; in particular, $p(G) = c(G)$ if $\Delta(G) = 1$.*

☆ A preliminary version is Nutov (2006) [16].

* Tel.: +972 9 778 1254.
E-mail address: nutov@openu.ac.il.

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.03.009

A simple connectivity requirement is when there should be a path from a root r to any other node. In this case, the min-cost variant is just the Min-Cost Arborescence problem which is solvable in polynomial time, while the Min-Power Arborescence problem is at least as hard as the Set-Cover problem; combined with the result of [19] this implies an $\Omega(\ln |V|)$-approximation threshold for the Min-Power Arborescence problem (namely, it cannot be approximated within $C \ln |V|$ for some universal constant $0 < C < 1$, unless P = NP). If we require a path from any node to the other, then we get the Min-Cost/Min-Power Strongly Connected Subgraph problem. The min-cost variant admits an easy 2-approximation algorithm, while the min-power variant is again Set-Cover hard.

A graph is *k-edge-outconnected from r* if it has k pairwise edge-disjoint rv-paths for every $v \in V$. A graph is *k-edge-connected* if it is k-edge-outconnected from every node. We consider the following generalization of the Min-Power Arborescence and the Min-Power Strongly Connected Subgraph problems studied in [2].

Min-Power k-Edge-Outconnected Subgraph
Instance: A (directed) graph $\mathcal{G} = (V, \mathcal{E})$ with edge costs $\{c(e) : e \in \mathcal{E}\}$, $r \in V$, and an integer k.
Objective: Find a min-power k-edge-outconnected from r spanning subgraph G of $\mathcal{G}$.

Min-Power k-Edge-Connected Subgraph
Instance: A (directed) graph $\mathcal{G} = (V, \mathcal{E})$ with edge costs $\{c(e) : e \in \mathcal{E}\}$ and an integer k.
Objective: Find a min-power k-edge-connected spanning subgraph G of $\mathcal{G}$.

Min-cost versions of these problems were studied extensively for both directed and undirected graphs; see surveys in [6,11,14]. Min-Cost k-Edge-Outconnected Subgraph is solvable in polynomial time [5]; see also more efficient algorithms in [8] and [7]. Min-Cost k-Edge-Connected Subgraph admits an easy 2-approximation algorithm, c.f., [14].

As intermediate problems, we consider the augmentation versions of the problems. Suppose that $\mathcal{G}$ has a subgraph $G_0 = (V, E_0)$ of power zero which is k_0-edge-outconnected from r, and the goal is to augment G_0 by a min-power edge set $F \subseteq \mathcal{E} - E_0$ so that the resulting graph $G = G_0 + F$ is k-edge-outconnected from r. Formally:

Min-Power (k_0, k)-Edge-Outconnectivity Augmentation
Instance: A graph $G_0 = (V, E_0)$ which is k_0-edge-outconnected from r, an edge set $\mathcal{I}$ on V with edge costs $\{c(e) : e \in \mathcal{I}\}$, and an integer $k > k_0$.
Objective: Find a min-power edge set $I \subseteq \mathcal{I}$ so that $G = G_0 + I$ is k-edge-outconnected from r.

In a similar way, the augmentation version Min-Power (k_0, k)-Edge-Connectivity Augmentation of Min-Power k-Edge-Connected Subgraph is defined. In [2] are given approximation algorithms for $k_0 = 0$ and $k = 1$: a $2H(n)$-approximation for Min-Power Arborescence and a $(2H(n)+1)$-approximation for Min-Power Strongly Connected Subgraph, where $n = |V|$ and $H(n)$ denotes the nth Harmonic number. Both problems generalize the Set-Cover problem (c.f., [2]), and thus the result in [2] is essentially tight up to a constant factor. For arbitrary k_0, k we prove:

Theorem 1.2. Min-Power (k_0, k)-Edge-Outconnectivity Augmentation *admits a* $3(k-k_0)H(n)$*-approximation algorithm.* Min-Power (k_0, k)-Edge-Connectivity Augmentation *admits a* $(k - k_0)(3H(n) + 1)$*-approximation algorithm. Thus each one of the problems* Min-Power k-Edge-Outconnected Subgraph *and* Min-Power k-Edge-Connected Subgraph *admits an* $O(k \ln n)$*-approximation algorithm.*

Remark. In the preliminary version [16] of this paper, the author claimed an $O(k \ln n)$-approximation algorithms also for the node-connectivity versions of the problems. However, the proof was found to contain an error by one of the referees of this paper.

The approximation ratio in Theorem 1.2 is $O(\ln n)$ for any fixed k, which is tight up to a constant factor if k is "small" (usually, $k \leq 3$ in practical networks), but may seem weak if k is large. However, it might be that a much better approximation algorithms do not exist. In [15] it is proved that for $k = \Omega(n^{\epsilon})$ the problems in Theorem 1.2 cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any fixed $\varepsilon > 0$, unless NP $\subseteq$ Quasi-P.

Theorem 1.2 is just an application of a general approximation algorithm for finding a min-power edge-cover of a certain widely studied type of set-families. We need some definitions to present this result.

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

- $\mathcal{F}$ is an intersecting family if $X \cap Y, X \cup Y \in \mathcal{F}$ for any $X, Y \in \mathcal{F}$ with $X \cap Y \neq \emptyset$.
- An edge set I *covers* $\mathcal{F}$ if for every $X \in \mathcal{F}$ there is an edge in I entering X, that is, there is $uv \in I$ with $u \in V - X$ and $v \in X$.

Min-Power Set-Family Edge-Cover
Instance: A set-family $\mathcal{F}$ on a groundset V and an edge set $\mathcal{I}$ on V with edge costs $\{c(e) : e \in \mathcal{I}\}$.
Objective: Find a minimum power $\mathcal{F}$-cover $I \subseteq \mathcal{I}$.

Given an instance of Min-Power Set-Family Edge-Cover we assume that $\mathcal{I}$ covers $\mathcal{F}$, as otherwise the problem has no feasible solution. We give a $3H(n)$-approximation algorithm for Min-Power Set-Family Edge-Cover with intersecting $\mathcal{F}$, but its polynomial implementation requires that certain queries related to $\mathcal{F}$ can be answered in polynomial time.

Definition 1.2. Given an edge set I on V (I is a partial cover of $\mathcal{F}$), the *residual family $\mathcal{F}_I$ of $\mathcal{F}$* (w.r.t. I) consists of all members of $\mathcal{F}$ that are uncovered by edges of I.

It is known that if $\mathcal{F}$ is intersecting, so is $\mathcal{F}_I$, for any I. Note that if $\mathcal{F}$ is intersecting, then for any $v, t \in V$, the family $\{X \in \mathcal{F}_I : t \in X, v \notin X\}$, if non-empty, is an intersecting family that has a unique inclusion minimal set; such a family is often called a *ring family* in the literature. For any edge set I on V and $v, t \in V$, make the following two assumptions (in our application they can be implemented using max-flow and min-cost k-flow algorithms).

Assumption 1. Computing the minimal member of the family $\{X \in \mathcal{F}_I : t \in X, v \notin X\}$ or determining that such does not exists can be done in polynomial time.

Assumption 2. Given an edge set with costs on V, a min-cost cover of the family $\{X \in \mathcal{F}_I : t \in X, v \notin X\}$ can be computed in polynomial time.

Theorem 1.3. Min-Power Set-Family Edge-Cover *with intersecting $\mathcal{F}$ admits a $3H(n)$-approximation algorithm under Assumptions 1 and 2.*

Theorem 1.3 is proved in Section 3. Here we show that Theorem 1.3 can be extended to so called "crossing families". A set-family $\mathcal{F}$ is a *crossing family* if $X \cap Y, X \cup Y \in \mathcal{F}$ for any $X, Y \in \mathcal{F}$ so that $X \cap Y, X - Y, Y - X, V - (X \cup Y)$ are all non-empty. Let us say that an edge set I is a *reverse cover* of $\mathcal{F}$ if for every $X \in \mathcal{F}$ there is an edge in I *leaving* X. It is known that (c.f., [7]):

Fact 1.4. *Let $\mathcal{F}$ be an intersecting family. If I is an inclusion minimal reverse cover of $\mathcal{F}$ then $d_I(v) \leq 1$ for every $v \in V$ (recall that $d_I(v)$ is the* outdegree *of v w.r.t. I), and thus the power of I equals it cost. In particular, I is a min-power reverse cover of $\mathcal{F}$ if, and only if, I is a min-cost reverse cover of $\mathcal{F}$.*

Any crossing family $\mathcal{F}$ can be naturally represented by two intersecting families as follows: fix $r \in V$ and define $\mathcal{F}_r^{in} = \{X \in \mathcal{F} : r \notin X\}$ and $\mathcal{F}_r^{out} = \{V - X : X \in \mathcal{F} - \mathcal{F}_r^{in}\}$. Then I covers $\mathcal{F}$ if, and only if, I is a cover of $\mathcal{F}_r^{in}$ and I is a reverse cover of $\mathcal{F}_r^{out}$. Combining with Fact 1.4, we get:

Corollary 1.5. *The problem of finding a min-power cover of a crossing family $\mathcal{F}$ on V admits a $(3H(n) + 1)$-approximation algorithm, if for some $r \in V$ Assumptions 1 and 2 are valid for $\mathcal{F}_r^{in}$, and if a min-cost reverse cover of $\mathcal{F}_r^{out}$ can be computed in polynomial time.*

A set-function f on 2^V is *positively intersecting supermodular* if $f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y)$ for any intersecting $X, Y \subset V$ with $f(X), f(Y) > 0$. An edge set I covers f if at least $f(X)$ edges in I enter every $X \subset V$. A $\{0, 1\}$-valued set-function f is positively intersecting supermodular if, and only if, its support family $\mathcal{F} = \{X \subseteq V : f(X) = 1\}$ is an intersecting family. A natural question is whether Theorem 1.3 extends to (positively) intersecting supermodular set-functions. As Min-Power k-Edge-Outconnected Subgraph is a particular case of the problem of finding a min-power cover of a positively intersecting supermodular set-function, such an extension is unlikely due to the hardness result of [15].

Our techniques are inspired by the algorithm of Klein and Ravi [12] for the undirected Node Weighted Steiner Forest problem. The Klein–Ravi algorithm [12] uses the "set-cover greedy approach" based on "density" considerations. At each step a "spider" (a subtree having at most one node of degree more than 2) is chosen that minimizes the ratio of spider's weight over the number of terminal pairs it connects minus 1. They proved that greedily adding spiders yields a $2H(n)$-approximation algorithm ($H(n)$ denotes the nth Harmonic number). For Min-Power Arborescence [2] gave a $2H(n)$-approximation algorithm using a similar method.

The main tool used to prove Theorem 1.3 is a decomposition of directed edge-covers of intersecting families into an analogue of spiders which we call "star-covers". This enables us to apply the Klein–Ravi [12] approach. However, "star-covers" are much more complicated than spiders, and the proof that any cover of an intersecting set-family can be properly decomposed into "star-covers" is substantially harder than the proof that every tree can be decomposed into spiders; see Section 2. Unlike [12], and other papers that used the approach of [12], e.g., [2], a star-cover is not necessarily a tree, and as we deal with covers of set-families, we cannot use specific graph properties. Another major difficulty is that inclusion minimal edge-covers of intersecting families can contain cycles. This is the reason why our approximation ratio is $3H(n)$, and not $2H(n)$ as in [12,2], where minimal feasible solutions are trees. (However, with some additional effort, it seems possible to improve the ratio in Theorem 1.3 to $(2 + \varepsilon)H(n)$; see a Remark at the end of Section 3.) Recently, based on the ideas of this paper, a more involved decomposition of *undirected* edge-covers was derived in [18] for so called "uncrossable" set-families, which are related to the undirected Node Weighted Steiner Network problem — a generalization of the Node Weighted Steiner Forest problem considered in [12].

This paper is organized as follows. In the rest of this section we introduce some notation used in the paper. Section 2 presents our decomposition of directed edge-covers of intersecting families. Theorems 1.3 and 1.2 are proved in Sections 3 and 4, respectively.

Notation: Let $G = (V, E)$ be a (directed) graph. For disjoint $X, Y \subseteq V$ let $\delta_G(X, Y) = \delta_E(X, Y)$ be the set of edges from X to Y in E. For brevity, $\delta_E(X) = \delta_E(X, V - X)$ is the set of edges in E leaving X, $d_E(X) = |\delta_E(X)|$, $\delta_E^{in}(X) = \delta_E(V - X, X)$ is the set of edges in E entering X, and $d_E^{in}(X) = |\delta_E^{in}(X)|$ is the indegree of X. Given edge costs $\{c(e) : e \in E\}$, the power of a node

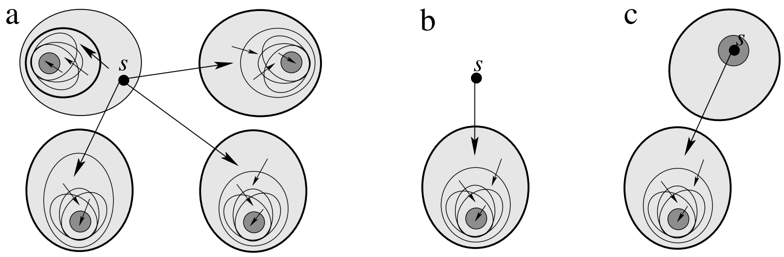

Fig. 1. Star-covers (min-cores are shown by dark gray circles).

v in G (with respect to c) is $p(v) = \max_{e \in \delta_E(v)} c(e)$, and the power of G is $p(G) = p_E(V) = \sum_{v \in V} p(v)$. For an edge set I, let $tails(I) = \{u : uv \in I\}$ denote the set of tails of the edges in I. Let $n = |V|$ and let opt denote the optimal solution value of a problem instance at hand.

2. Decomposition of covers of intersecting families

We start by describing some simple properties of intersecting families.

Definition 2.1. A member of a set-family $\mathcal{F}$ is an $\mathcal{F}$*-core* if it does not contain two disjoint members of $\mathcal{F}$; an inclusion minimal $\mathcal{F}$-core is a *min-*$\mathcal{F}$*-core* and an inclusion maximal $\mathcal{F}$-core is a *max-*$\mathcal{F}$*-core*. Let $\mathcal{C}(\mathcal{F})$ denote the family of min-$\mathcal{F}$-cores, and let $\mathcal{M}(\mathcal{F})$ denote the family of max-$\mathcal{F}$-cores. We will often use core, min-core, and max-core, instead of $\mathcal{F}$-core, min-$\mathcal{F}$-core, and max-$\mathcal{F}$-core, respectively, if $\mathcal{F}$ is understood.

Fact 2.1. *Let $\mathcal{F}$ be an intersecting family. Then the family of $\mathcal{F}$-cores is also an intersecting family. Consequently, the members of each one of the families $\mathcal{C}(\mathcal{F})$ and $\mathcal{M}(\mathcal{F})$ are pairwise disjoint, and for every min-core C there is a unique max-core M_C containing C.*

Proof. Clearly, the min-cores are just the inclusion minimal members of $\mathcal{F}$. Note that if $X \cap C \neq \emptyset$ for $X \in \mathcal{F}$ and $C \in \mathcal{C}(\mathcal{F})$ then $C \subseteq X$. Indeed, $X \cap C \in \mathcal{F}$ since $\mathcal{F}$ is an intersecting family; as C is a min-core, we must have $X \cap C = C$, implying $C \subseteq X$. We prove that if X, Y are cores with $X \cap Y \neq \emptyset$ then $X \cap Y, X \cup Y$ are also cores. We have $X \cap Y, X \cup Y \in \mathcal{F}$, since $\mathcal{F}$ is an intersecting family. In particular, $X \cap Y$ contains some min-core $C \in \mathcal{C}(\mathcal{F})$. Since X, Y are cores, none of them contains a min-core disjoint to C. Consequently, each one of $X \cap Y, X \cup Y$ belongs to $\mathcal{F}$ and does not intersect a min-core disjoint to C. Hence $X \cap Y, X \cup Y$ are cores, as claimed. □

We now describe a variant of the decomposition of [12] of a directed tree (arborescence) into spiders.

Definition 2.2. A *spider* is a directed tree with at least one edge and at most one node of outdegree ≥ 2. Given a subset U of nodes of a directed tree T, a collection $\mathcal{S} = \{S_1, \ldots, S_q\}$ of spiders contained in T is a *spider decomposition of* T, U if $tails(S_i) \cap tails(S_j) = \emptyset$ for all $i \neq j = 1, \ldots, q$, and if every $u \in U$ is covered by (that is, belongs to) at most one member of $\mathcal{S}$.

Lemma 2.2 (*[12]*). *Any subset U of nodes of a directed tree (or of a directed forest) T can be covered by a spider decomposition.*

Proof-Sketch. By induction on U. We may assume that every leaf of T belongs to U. If $|U| = 1$ the statement is trivial. Otherwise, T has a node s so that the subtree S that consists of s and all its descendants is a spider, and for every $u \in U \cap S$, either $u = s$ or u is a leaf of S. Now, if $U \subseteq S$, then we are done. Otherwise, let $T' \leftarrow T - S$ and $U' \leftarrow U - (S \cap U)$. By the induction hypothesis, T' admits a spider decomposition $\mathcal{S}'$ that covers all U'. It is not hard to verify that then $\mathcal{S}' \cup \{S\}$ is a spider decomposition of T that covers all U. □

For $C \in \mathcal{C}(\mathcal{F})$ and $s \in V$ let

$$\mathcal{F}(s, C) = \{X \in \mathcal{F} : C \subseteq X \subseteq M_C, s \notin X\}$$

denote the family of $\mathcal{F}$-cores that contain C and do not contain s. Given $s \in V$ we will say that an edge set S *star-covers* (from s) a min-core $C \in \mathcal{C}(\mathcal{F})$ if S covers the family $\mathcal{F}(s, C)$. Note that if $s \in C$ then $\mathcal{F}(s, C) = \emptyset$; in this case $S = \emptyset$ star-covers C, although no edge in S covers C itself.

For directed covers of intersecting set-families, we define the following analogue of spiders:

Definition 2.3. Let $\mathcal{F}$ be an intersecting set-family on V. An edge set S on V is a *star-cover* (with center s) of a subfamily $\mathcal{C} \subseteq \mathcal{C}(\mathcal{F})$ of min-cores if (see Fig. 1) S can be partitioned into $\mathcal{F}(s, C)$-covers $\{S_C : C \in \mathcal{C}\}$ (possibly $S_C = \emptyset$ for one $C \in \mathcal{C}$, if $s \in C$), such that if $\mathcal{C} = \{C\}$ then $s \notin M_C$, and such that for every $C \in \mathcal{C}$:

- Edges in $S_C - \delta_S(s)$ have their both endnodes in M_C, and no two such edges share a tail.
- If $s \notin M_C$, then S_C contains a unique edge e_{M_C} from s to M_C.

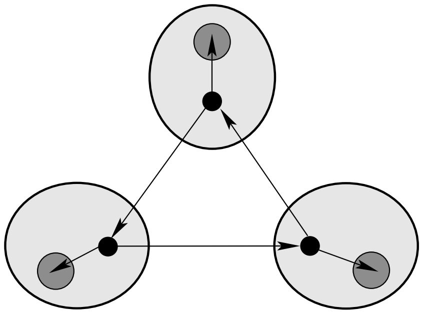

Fig. 2. An example showing that the bound $\lceil 2|\mathcal{C}(\mathcal{F})|/3\rceil$ in Theorem 2.3 is tight.

We now state our definition of "star-cover decomposition" of directed covers of intersecting set-families.

Definition 2.4. Let I be an $\mathcal{F}$-cover of an intersecting set-family $\mathcal{F}$ on V. A collection $\mathcal{S} = \{S_1, \ldots, S_q\}$ of star-covers is a *star-cover decomposition of* I if $tails(S_i) \cap tails(S_j) = \emptyset$ for all $i \neq j = 1, \ldots, q$, and if every $C \in \mathcal{C}(\mathcal{F})$ is star-covered by at most one member of $\mathcal{S}$.

The main technical result of this paper is the following "analogue" of Lemma 2.2:

Theorem 2.3 (*The Star-Cover Decomposition Theorem*). *Any directed cover I of an intersecting family $\mathcal{F}$ admits a star-cover decomposition that star-covers at least $\lceil 2|\mathcal{C}(\mathcal{F})|/3\rceil$ min-cores.*

Example. The bound $\lceil 2|\mathcal{C}(\mathcal{F})|/3\rceil$ in Theorem 2.3 is tight even for laminar set-families; see Fig. 2. In this example, there are three distinct star-covers each star-covering two min-cores, but there is no star-cover decomposition that star-covers all min-cores.

The proof of Theorem 2.3 follows. In what follows, let $\mathcal{F}$ be an intersecting family and let I be an *inclusion minimal* $\mathcal{F}$-cover. We need to establish some properties of I. By the minimality of I, for every $e \in I$ there exists $W_e \in \mathcal{F}$ such that $\delta_I^{in}(W_e) = \{e\}$; we call such W_e a *witness set for* e; note that e might have several distinct witness sets.

Lemma 2.4. *Let W_e, W_f be witness sets of two distinct edges $e, f \in I$ so that $W_e \cap W_f \neq \emptyset$. Then $W_e \cap W_f$ is a witness set for one of e, f and $W_e \cup W_f$ is a witness set for the other.*

Proof. Note that there is an edge in I entering $W_e \cap W_f$ and there is an edge in I entering $W_e \cup W_f$; this is since $W_e, W_f \in \mathcal{F}$ and $W_e \cap W_f \neq \emptyset$ implies that $W_e \cap W_f$, $W_e \cup W_f$ belong to $\mathcal{F}$ and thus each of them is covered by some edge in I. However, if for arbitrary intersecting sets X, Y an edge covers at least one of $X \cap Y, X \cup Y$ then it also covers at least one of X, Y, and if some edge covers both $X \cap Y$ and $X \cup Y$ then it must cover both X and Y. Thus no edge in $I - \{e, f\}$ can cover $W_e \cap W_f$ or $W_e \cup W_f$, so one of e, f covers $W_e \cap W_f$, and thus the other must cover $W_e \cup W_f$. $\square$

Corollary 2.5. $d_I^{in}(C) = 1$ *for any min-$\mathcal{F}$-core C.*

Proof. Clearly, $\delta_I^{in}(C) \geq 1$ for any min-$\mathcal{F}$-core C, since I is an $\mathcal{F}$-cover and $C \in \mathcal{F}$. Assume to the contrary that there are distinct $e, f \in \delta_I^{in}(C)$ for some min-core C; let W_e, W_f be their witness sets. As the head of each of e, f is in C, we must have $W_e \cap C \neq \emptyset$ and $W_f \cap C \neq \emptyset$. Then $C \subseteq W_e \cap W_f$ by Fact 2.1, so $W_e \cap W_f \neq \emptyset$, and $e, f \in \delta_I^{in}(W_e \cap W_f)$. This contradicts Lemma 2.4. $\square$

An intersecting family $\mathcal{F}$ is *simple* if every member of $\mathcal{F}$ is an $\mathcal{F}$-core. It would be sufficient to prove Theorem 2.3 for simple families. If $\mathcal{F}$ is not simple, we may replace $\mathcal{F}$ by the family of $\mathcal{F}$-cores; by Fact 2.1, the latter is intersecting if $\mathcal{F}$ is. So throughout the rest of this section, assume that $\mathcal{F}$ is a simple intersecting family.

Lemma 2.6. *Let $M \in \mathcal{M}(\mathcal{F})$ and let $I(M) = \{uv \in I : u, v \in M\}$. Assuming $I(M) \neq \emptyset$, there exists a unique ordering $e_1, e_2, \ldots, e_q$ of $I(M)$ and a nested family $X_1 \subset X_2 \cdots \subset X_q \subset M$ of sets in $\mathcal{F}$ so that: X_{j+1} is a min-core of $\mathcal{F}_{I_j}$ where $I_j = \{e_1, \ldots, e_j\}$ (and $I_0 = \emptyset$), and e_j is the unique edge in I entering X_j. Furthermore, for any j, if s is a tail of e_j and C is the min-core contained in M, then I_j is an $\mathcal{F}(s, C)$-cover.*

Proof. Let X_1 be the min-core contained in M. By Corollary 2.5 there is a unique edge in I entering X_1, say e_1. Suppose that e_1 covers M. We claim that then $I(M) = \emptyset$, so the statement holds in this case. Suppose to the contrary that there is $e \in I(M)$, and let W_e be a witness set for e (such exists, by the minimality of I). By Fact 2.1, and since we assume that $\mathcal{F}$ is simple, we must have $X_1 \subseteq W_e \subseteq M$. However, e_1 covers all cores contained in M, hence e_1 covers W_e. This contradicts that W_e is a witness set for e.

If e_1 does not cover M, let X_2 be the minimal $\mathcal{F}_{e_1}$-core contained in M. By Fact 2.1, $X_1 \subset X_2$. Let e_2 be the unique edge in I entering X_2, and so on, until M is covered by some edge e_{q+1}. In such a way we obtain sequences $e_1, e_2, \ldots, e_q$ of edges in $I(M)$ (an additional edge $e_{q+1} \notin I(M)$ since it enters M), and $X_1 \subset X_2 \cdots \subset X_q \subset M$ of sets in $\mathcal{F}$ so that: X_{j+1} is the core of $\mathcal{F}_{I_j}$, where $I_j = \{e_1, \ldots, e_j\}$ and e_j is the unique edge in I entering X_j. The statement follows. $\square$

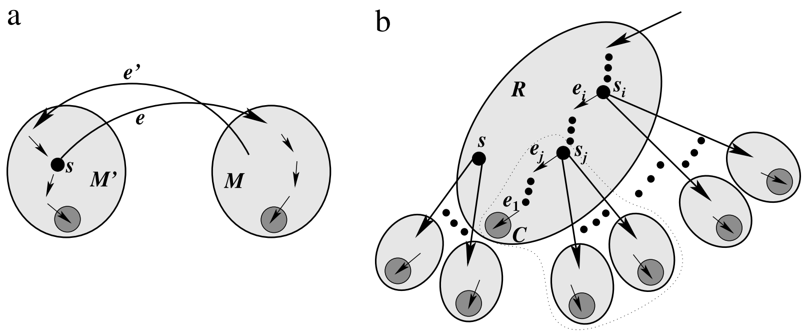

Fig. 3. Illustration to the proof of Lemma 2.8. (a) J is a 2-cycle. (b) J is a directed tree.

Remark. Lemma 2.6 is not true if $\mathcal{F}$ is not simple. A counterexample is: $V = \{u, v, x, y, z, a\}$, $\mathcal{F} = \{\{v\}, \{a\}, \{v, z\}, \{v, z, y\}, \{v, z, u, a\}, \{v, z, u, a, y\}\}$ and $I = \{uv, ua, xy, yz\}$. Then $\mathcal{F}$ is intersecting but not simple — $\{v, z, u, a\}$ contains the two min-cores $\{v\}$ and $\{a\}$. The edge $e_1 = uv$ covers both the max-core $M = \{v, z, y\}$ and the min-core $\{v\}$, but $I(M) = \{yz\} \neq \emptyset$. Also, I is a minimal cover, as the witness sets are: $\{v\}$ for uv, $\{a\}$ for ua, $\{v, z, u, a\}$ for yz, and $\{v, z, u, a, y\}$ for xy.

Corollary 2.7. *Let $M \in \mathcal{M}(\mathcal{F})$. Then $d_{I(M)}(v) \leq 1$ for every $v \in M$; thus $p(I(M)) = c(I(M))$, namely, the power of $I(M)$ equals its cost. Furthermore, let e_M be the unique edge in I entering the minimal core X of $\mathcal{F}_{I(M)}$ contained in M (possibly $X = M$). Then $\delta_I^{in}(M) = \{e_M\}$ and $I(M) + e_M$ covers the family $\{Y \in \mathcal{F} : Y \subseteq M\}$.*

Proof. The first statement follows from Lemma 2.6 and the last statement of Proposition 1.1. It is also easy to see that $I(M) + e_M$ covers the family $\{Y \in \mathcal{F} : Y \subseteq M\}$. We prove that $\delta_I^{in}(M) = \{e_M\}$. Suppose to the contrary that there is $f \in I - \{e_M\}$ entering M. By Corollary 2.5, X is a witness set for e_M. Let W_f be a witness set for f. The head of f is in $W_f \cap M$. Thus $W_f \subseteq M$, by Fact 2.1 and since $\mathcal{F}$ is simple. As M is a core, $X \cap W_f \neq \emptyset$. But then both e_M and f enter $X \cup W_f \subseteq M$, contradicting Lemma 2.4. □

For $M \in \mathcal{M}(\mathcal{F})$ let e_M be the (unique, by Corollary 2.7) edge in I entering M, and let $I_{\mathcal{M}} = \{e_M : M \in \mathcal{M}(\mathcal{F})\}$. By Fact 2.1, the members of $\mathcal{M}(\mathcal{F})$ are pairwise disjoint. Obtain an auxiliary graph J from $(V, I_{\mathcal{M}})$ by shrinking every $M \in \mathcal{M}(\mathcal{F})$ into a single node, which we call *compound node*. As every edge e_M has its head in M and its tail not in M, the edge set of J is exactly $I_{\mathcal{M}}$. In J, the indegree of every node is at most 1, by Corollary 2.7. Furthermore, every node of indegree 1 is a compound node. Thus, ignoring the isolated nodes, J is a collection of node-disjoint graphs ("components") of the following type: each component is either a cycle, or a directed tree rooted at a non-compound node, or a cycle with disjoint directed trees attached to it by the roots.

Note that any node of a cycle and any leaf of J is a compound node, since its indegree is 1. Thus if a component J_q of J contains a cycle, then either J_q is a 2-cycle, or J_q has at least 3 compound nodes. In the latter case, delete one edge e_M from the cycle, remove from $\mathcal{F}$ all the members of $\mathcal{F}(M) = \{X \in \mathcal{F} : X \subseteq M\}$, and remove from I the set $I(M)$. Note that at least $\lceil 2|\mathcal{C}(\mathcal{F})|/3\rceil$ max-cores remain. Thus we obtain a new simple intersecting family $\mathcal{F}$ and a new $\mathcal{F}$-cover I, so that every component of the new graph J will be either a 2-cycle or a tree. If we can show that in this case I admits a star-cover decomposition that star-covers all min-cores, then we are done. This can be proved for each component J_q of J separately. More formally, let $\mathcal{M}_q$ be the set of max-cores corresponding to the compound nodes of J_q, let $\mathcal{F}_q = \{X \in \mathcal{F} : X \subseteq M, M \in \mathcal{M}_q\}$, and let $I_q = \{uv \in I : v \in M \cap T, M \in \mathcal{M}_q\}$. Note that I_q covers $\mathcal{F}_q$, since I is an $\mathcal{F}$-cover, and since no edge in $I - I_q$ can cover a member of $\mathcal{F}_q$. The families $\mathcal{M}_q$ partition $\mathcal{M}$, the families $\mathcal{F}_q$ partition $\mathcal{F}$, and the families I_q partition I. Hence if for every q we obtain a star-cover decomposition $\mathcal{S}_q$ of I_q that star-covers $\mathcal{C}(\mathcal{F}_q)$, then the union of the star-covers $\mathcal{S}_q$ is a star-cover decomposition of I that star-covers $\mathcal{C}(\mathcal{F})$. Consequently, the following statement implies Theorem 2.3:

Lemma 2.8. *Let I be a minimal cover of a simple intersecting family $\mathcal{F}$ on V. Let J be the graph obtained from $(V, I_{\mathcal{M}})$ by shrinking every $M \in \mathcal{M}(\mathcal{F})$ into a compound node. If J is a 2-cycle or if J is a directed tree, then I admits a star-cover decomposition that star-covers all min-cores.*

Proof. Suppose that J is a 2-cycle connecting $M, M' \in \mathcal{M}(\mathcal{F})$ (see Fig. 3(a) for illustration). Let $e = e_M$ be the edge of this cycle entering M and let $e' = e_{M'}$ be the edge of this cycle entering M'. Let C and C' be the min-cores contained in M and M', respectively. Since $M \cap M' = \emptyset$, $tails(I(M)) \cap tails(I(M')) = \emptyset$. Note that $I(M) \cup \{e\}$ star-covers C and $I(M') \cup \{e'\}$ star-covers C', by the last statement in Corollary 2.7. Hence if both the tail of e is not in $tails(I(M'))$ and the tail of e' is not in $tails(I(M))$, then $\{I(M) \cup \{e\}, I(M') \cup \{e'\}\}$ is a star-cover decomposition as required. Otherwise, the tail of e is in $tails(I(M'))$ or the tail of e' is in $tails(I(M))$, and suppose that the former holds (see Fig. 3(a)); the proof of the latter case is identical. Let s be the tail of e. Then a star-cover that star-covers both C and C' is obtained by adding to $I(M) \cup \{e\}$ any $\mathcal{F}(s, C')$-cover contained in $I(M')$; such $\mathcal{F}(s, C')$-cover exists, by the last statement in Lemma 2.6.

Let us now consider the case when J is a tree with root r. Every node of J distinct from r is a compound node. We prove the statement by induction on the number of compound nodes in J, using a similar approach as in the proof of Lemma 2.2. The induction base is when J has a unique compound node corresponding to a max-core M. Then $S = I(M) \cup \{e_M\}$ is a star-cover with center $s = r$, of the type in Fig. 1(b). Otherwise, J has a node (a father of the farthest node from r) so that all its children are leaves. If this node is r, then I is a star-cover with center r of $\mathcal{C}(\mathcal{F})$ and we are done. Otherwise, this node is a compound node that corresponds to a max-core R (see Fig. 3(b)). Let $Y = tails(\delta(R))$ be the set of tails of the edges in J leaving R, and let $Z = tails(I(R))$ be the set of tails of edges in $I(R)$. Consider two cases: $Y - Z \neq \emptyset$ and $Y \subseteq Z$.

If there is $s \in Y - Z$ (see Fig. 3(b)), then we form the star-cover $S = \bigcup\{\{e_M\} \cup I(M) : e_M \in \delta_J(s)\}$ by taking every edge $e_M \in \delta(s)$ leaving s together with the edges with both endpoints in M. By Corollary 2.7, S covers the family $\mathcal{F}(M) = \{X \in \mathcal{F} : X \subseteq M\}$ for every $M \in \mathcal{M}(\mathcal{F})$ so that $e_M \in \delta(s)$. Now, set $I' \leftarrow I - S$ and $\mathcal{F}' \leftarrow \mathcal{F} - \bigcup_{e_M \in \delta(s)} \mathcal{F}(M)$ (so $\mathcal{F}'$ is obtained by removing from $\mathcal{F}$ the cores containing a min-core star-covered by S). Since $\mathcal{F}$ is simple, $\mathcal{F}'$ is also an intersecting family that is simple (note that no new cores appear in $\mathcal{F}'$). The residual instance $\mathcal{F}', I'$ satisfies the assumptions of the lemma, and thus, by the induction hypothesis, admits a star-cover decomposition $\mathcal{S}'$. Since no edge in S has a tail in common with an edge in $I' = I - S$, $\mathcal{S}' \cup \{S\}$ is a star-cover decomposition of I as required.

Now suppose that $Y \subseteq Z$. Let $e_1, \ldots, e_q$ be an ordering of the edges in $I(R)$ as in Lemma 2.6, and let s_i be the tail of e_i. For every $s_i \in Y$ we form a star-cover S_i as in the previous case, by taking every edge $e_M \in \delta(s_i)$ leaving s_i together with the edges with both endpoints in M. This gives a star-cover decomposition that star-covers all min-cores contained in the children of R. We will extend it to star-cover the min-core C contained in R as follows. Let j be the least index so that $s_j \in Y$ and let $S_C = \{e_j, e_{j-1}, \ldots, e_1\}$ (see Fig. 3(b)). Note that $tails(S_C) \cap tails(S_i) = \emptyset$ for $s_i \in Y \setminus \{s_j\}$ while $tails(S_C) \cap tails(S_j) = \{s_j\}$. By Lemma 2.6, S_C is an (s_j, C)-cover, so $S_j \cup S_C$ also star-covers C. Thus we obtain a star-cover decomposition that covers all min-cores contained in the children of R and also the core C contained in R. We then remove from I all edges taken into the decomposition, and remove from $\mathcal{F}$ the cores containing a min-core star-covered by the decomposition. Note that R remains a compound node and, in the terminology of Lemma 2.6, the min-core inside R is X_{j+1}. Now we apply the inductive hypothesis in the same way as in the case $s \in Y - Z$. □

The proof of Theorem 2.3 is now complete.

Remark. As far as we can see, the bound $\lceil 2|\mathcal{C}(\mathcal{F})|/3 \rceil$ in Theorem 2.3 can be improved to $|\mathcal{C}(\mathcal{F})|$ minus the number of odd cycles in J. Specifically, one can prove that if J is an even cycle, then all its cores can be star-covered by the decomposition, in a similar way as for 2-cycles; the same is valid if J contains an even cycle, by combining the proof for even cycles and trees. Components containing an odd cycle, are handled in the same way as in the proof, by removing one edge from the cycle, and give up star-covering the core entered by the removed edge.

3. Covering intersecting families (Proof of Theorem 1.3)

We use a well-known result about the performance of a greedy algorithm for the following type of "covering problems":

Covering Problem
Instance: Set-functions ν, p on groundset $\mathcal{I}$ given by an evaluation oracle, so that ν is integral and $\nu(\mathcal{I}) = 0$.
Objective: Find $I \subseteq \mathcal{I}$ with $\nu(I) = 0$ and with $p(I)$ minimized.

Definition 3.1. A set-function f on $2^{\mathcal{I}}$ is:
- *decreasing* (resp, *increasing*) if $f(I_2) \leq f(I_1)$ (resp., if $f(I_2) \geq f(I_1)$) for any $I_1 \subset I_2 \subseteq E$.
- *subadditive* if $f(I_1 \cup I_2) \leq f(I_1) + f(I_2)$ for all $I_1, I_2 \subseteq E$.

In the Covering Problem, ν is the *deficiency function* (it is assumed to be decreasing and measures how far is I from being a feasible solution) and p the *payment function* (assumed to be increasing and subadditive). In our case, p is just the power function, and $\nu(I)$ is the number of minimal cores in $\mathcal{F}_I$. Let $\rho > 1$ and let opt be the optimal solution value for the Covering Problem. The ρ-*Greedy Algorithm* starts with $I = \emptyset$ and iteratively adds subsets of $\mathcal{I} - I$ to I one after the other using the following rule. As long as $\nu(I) \geq 1$ it adds to I a set $S \subseteq \mathcal{I} - I$ so that

$$\sigma_I(S) = \frac{p(S)}{\nu(I) - \nu(I + S)} \leq \rho \cdot \frac{\mathsf{opt}}{\nu(I)}. \tag{1}$$

$\sigma_I(S)$ is called the *density* of S. The following statement is known, c.f., [12].

Theorem 3.1. *For any* Covering Problem *with ν decreasing and p increasing and subadditive, the ρ-Approximate Greedy Algorithm computes a solution I with $p(I) \leq \rho H(\nu(\emptyset)) \cdot$ opt.*

In the rest of this section we prove the following statement:

Lemma 3.2. *Let $\nu(I)$ be the number of minimal cores in $\mathcal{F}_I$. Then an edge set $S \subseteq \mathcal{I} - I$ satisfying (1) with $\rho = 3$ can be found in polynomial time under Assumptions 1 and 2.*

For simplicity of exposition, let us revise our notation and set $\mathcal{F} \leftarrow \mathcal{F}_I$ and $\mathcal{I} \leftarrow \mathcal{I} - I$. Let $\nu = \nu(\emptyset)$. Then we need to show that under Assumptions 1 and 2 one can find in polynomial time an edge set $S \subseteq \mathcal{I}$ so that:

$$\sigma(S) = \frac{p(S)}{\nu - \nu(S)} \leq 3 \cdot \frac{\mathsf{opt}}{\nu}. \tag{2}$$

Lemma 3.3. *If S star-covers d min-cores then $\nu - \nu(S) \geq \Delta(S) \geq d/2$, where $\Delta(S) = \max\{1, d-1\}$.*

Proof. Clearly, min-$\mathcal{F}$-cores not star-covered by S remain min-cores of $\mathcal{F}_S$. Any other min-$\mathcal{F}_S$-core X must contain a min-$\mathcal{F}$-core C, and C is star-covered by S. In particular, S covers every member of $\mathcal{F}(s, C)$, where s is the center of S, and $s \notin M_C$ if $d = 1$. We claim that $s \in X$ and thus:
- If $d = 1$ then no such X exists. Otherwise, since $s \notin M_C$, S covers every $\mathcal{F}$-core containing C; hence X must contain a min-$\mathcal{F}$-core $C' \neq C$, by Fact 2.1. But then C', and not X, is a min-$\mathcal{F}_S$-core.
- If $d \geq 2$ then there is at most one such X. This is since the min-$\mathcal{F}_S$-cores are pairwise disjoint.

Suppose to the contrary that $s \notin X$. Then X contains another min-$\mathcal{F}$-core $C' \neq C$; otherwise, $X \in \mathcal{F}(s, C)$, but S covers $\mathcal{F}(s, C)$, contradicting that $X \in \mathcal{F}_S$. Consequently, $s \notin M_C$ or $s \notin M_{C'}$, say $s \notin M_C$ (note that C' is also star-covered by S since X and not C' is a min-core of $\mathcal{F}_S$, hence we can interchange the roles of C and C'). Let $Y = X \cap M_C$. Then $Y \in \mathcal{F}$ and $Y \subseteq M_C$, thus S covers Y, since S covers all $\mathcal{F}$-cores contained in M_C. Consequently, there is an edge $uv \in S$ entering Y. Since uv does not cover X, we must have $u \in X - M_C$. But then uv covers M_C, implying, by the definition of a star-cover, that $u = s$. □

Lemma 3.4. *There exists a star-cover $S \subseteq \mathcal{I}$ so that $p(S)/\Delta(S) \leq 3 \cdot \mathsf{opt}/\nu$.*

Proof. Let $I \subseteq \mathcal{I}$ be an optimal $\mathcal{F}$-cover, so $p(I) = \mathsf{opt}$. By Theorem 2.3, I admits a star-cover decomposition $S_1, \ldots, S_t$ that star-covers at least $2|\mathcal{C}(\mathcal{F})|/3 = 2\nu/3$ min-cores. Recalling that every min-$\mathcal{F}$-core is star-covered at most once, the statement follows by a simple averaging argument. Let $p_i = p(S_i)$, let d_i be the number of min-$\mathcal{F}$-cores star-covered by S_i, and let $\Delta_i = \Delta(S_i) \geq d_i/2$. We have $\sum_{i=1}^{t} p_i \leq p(I) = \mathsf{opt}$ and $\sum_{i=1}^{t} \Delta_i \geq \frac{1}{2}\sum_{i=1}^{t} d_i \geq \frac{1}{2} \cdot \frac{2}{3}\nu \geq \nu/3$. Thus

$$\frac{\sum_{i=1}^{t} p_i}{\sum_{i=1}^{t} \Delta_i} \leq \frac{p(I)}{\nu/3} = 3 \cdot \frac{p(I)}{\nu}.$$

Consequently, there must be an index i so that $p_i/\Delta_i \leq 3 \cdot p(I)/\nu = 3 \cdot \mathsf{opt}/\nu$. □

Note that by Lemma 3.3, (2) holds for any star-cover S as in Lemma 3.4. Now we will show how to find such S in polynomial time, under Assumptions 1 and 2.

Lemma 3.5. *Let $\mathcal{F}$ be an intersecting family on V and let $\mathcal{I}$ be an edge set with costs on V. Then the following can be computed in polynomial time under Assumptions 1 and 2:*

(i) *The families $\mathcal{C}(\mathcal{F})$ and $\mathcal{M}(\mathcal{F})$.*
(ii) *A min-cost cover of the family $\mathcal{F}(v, C)$ for any $C \in \mathcal{C}(\mathcal{F})$ and $v \in V$.*

Proof. The family $\mathcal{C}(\mathcal{F})$ of min-$\mathcal{F}$-cores can be computed as follows. For every $v, t \in V$ compute the minimal member C_{vt} of the ring family $\{X \in \mathcal{F} : t \in X, v \notin X\}$, or determine that such does not exist, using the algorithm as in Assumption 1. The inclusion minimal (non-empty) members among the sets C_{vt} computed are the min-$\mathcal{F}$-cores.

After the min-$\mathcal{F}$-cores are found, to find the family $\mathcal{M}(\mathcal{F})$ of max-$\mathcal{F}$-cores, for every min-core $C \in \mathcal{C}(\mathcal{F})$ we show a decision procedure that checks if a given node $u \in V - C$ is in $M_C \in \mathcal{M}(\mathcal{F})$. Thus to find M_C it is enough to check every node. Our decision procedure is as follows. We fix some $t \in C$. The procedure accepts u if there exists $v \in V$ so that the minimal member of the ring family $\{X \in \mathcal{F}_{\{ut\}} : t \in X, v \notin X\}$ contains C and does not contain any other min-$\mathcal{F}$-core (this can be checked in polynomial time, by Assumption 1). Indeed, if $u \in M_C$ then the procedure accepts u for any $v \in V - M_C$; otherwise, if $u \notin M_C$, then u is not accepted, since any member of $\mathcal{F}_{ut}$ must contain a min-core distinct from X, by Fact 2.1.

Finally, we will show how to find a min-cost $\mathcal{F}(v, C)$-cover for given $C \in \mathcal{C}(\mathcal{F})$ and $v \in V$. Let J be an edge set of a star with center v and an edge from v to every min-core distinct from C that does not contain v ($J = \emptyset$ if $\mathcal{C}(\mathcal{F}) = \{C\}$). It is not hard to verify that for any $t \in C$, $\mathcal{F}(v, C) = \{X \in \mathcal{F}_J : t \in X, v \notin X\}$. Hence to find a min-cost $\mathcal{F}(v, C)$-cover, we find a min-cost cover of the family $\{X \in \mathcal{F}_J : t \in X, v \notin X\}$, which can be implemented in polynomial time, by Assumption 2. □

Lemma 3.6. *A star-cover $S \subseteq \mathcal{I}$ that minimizes $p(S)/\Delta(S)$ can be found in polynomial time under Assumptions 1 and 2.*

Proof. First compute the families $\mathcal{C}(\mathcal{F})$ and $\mathcal{M}(\mathcal{F})$; this can be implemented in polynomial time, by Lemma 3.5. Second, for every $v \in M_C$ for some $C \in \mathcal{C}(\mathcal{F})$ define the weight $w(v)$ of v to be the minimum cost of an $\mathcal{F}(v, C)$-cover among the edges in $\mathcal{I}$ with both endpoints in M_C, if such exists, and $w(v) = \infty$ otherwise; this also can be implemented in polynomial time, by Lemma 3.5. Assume that we know the center s and its power $p(s) = p_S(s)$ in S; there are $O(n^2)$ distinct choices. Among the edges leaving s, delete all edges of cost $> p(s)$, and zero the costs of the others. Construct an auxiliary weighted star T with center s as follows. For every min-core $C \in \mathcal{C}(\mathcal{F})$ add a node v_C and an edge from s to v_C. The weight $W(v_C)$ of v_C is defined by:
- If $s \notin M_C$, then $W(v_C) = \min\{w(v) : v \in M_C, sv \in \mathcal{I}, c(sv) \leq p(s)\}$ is the minimum weight of a neighbor of s contained in M_C, if such exists, and $W(v_C) = \infty$ otherwise.
- If $s \in M_{C'}$ for some $C' \in \mathcal{C}(\mathcal{F})$ then $W(v_{C'})$ is the minimum cost of an $\mathcal{F}(s, C')$-cover, if such exists (in particular, $W(v_{C'}) = 0$ if $\mathcal{F}(s, C') = \emptyset$), and $W(v_{C'}) = \infty$ otherwise.

We now see that our goal is to compute a sub-star S of T, $S \neq \{sv_{C'}\}$ if $s \in M_{C'}$, that minimizes $W(S)/\max\{|L_S| - 1, 1\}$, where $W(S) = p(s) + W(L_S)$ and L_S is the set of leaves of S. Sort the leaves of T distinct from $v_{C'}$ by increasing weight, say $W(v_1) \leq W(v_2) \leq \ldots \leq W(v_q)$. Assuming that $sv_{C'} \notin S$, let $\sigma_1 = p(s) + W(v_1)$ and $\sigma_j = (p(s) + \sum_{i=1}^{j} W(v_i))/(j-1)$, $j = 2, \ldots, q$. Assuming that $sv_{C'} \in S$, let $\sigma_j = (p(s) + w(v_{C'}) + \sum_{i=1}^{j} W(v_i))/j$, $j = 1, \ldots, q$. In both cases, we can find the index j for which σ_j is minimum, which determines the required star-cover. $\square$

The proof of Lemma 3.2, and thus also of Theorem 1.3 is complete.

Remark. As far as we can see, the approximation ratio in Theorem 1.3 can be improved to $(2+\varepsilon)H(n)$ if, in addition to star-covers, we will also consider "ℓ-cycle-covers"; ℓ-cycle-cover Q is obtained by taking a cycle of length $\leq \ell$ on max-cores, and adding an $\mathcal{F}(v, C)$-cover contained in M_C for every cycle edge uv entering a max-core M_C. An analogue of Theorem 2.3 would state that any cover of an intersecting family $\mathcal{F}$ admits a decomposition into star-covers and ℓ-cycle-covers that covers at least $(\ell+1)|\mathcal{C}(\mathcal{F})|/(\ell+2)$ min-cores. An analogue of Lemma 3.3 would state that $\nu - \nu(Q) \geq \ell - 1$, if Q is obtained from a cycle of length ℓ; hence $\sigma(Q) \leq p(Q)/(\ell - 1)$ for such Q. For any ℓ, we can find an ℓ-cycle-cover Q minimizing $\sigma(Q)$ in time $n^{q(\ell)}$, where $q(\ell)$ is polynomial in ℓ (details omitted). Setting $\ell = \lfloor 1/\varepsilon \rfloor$, we obtain a $(2+\varepsilon)H(n)$-approximation scheme.

4. Proof of Theorem 1.2

We give a $3H(n)$-approximation algorithm for Min-Power $(\ell, \ell+1)$-Edge-Outconnectivity Augmentation, that is, for the problems of finding a min-power augmenting edge set that increases the edge-outconnectivity from r by 1. We apply this algorithm sequentially for $\ell = k_0, \ldots, k-1$ to produce edge sets $I_{k_0}, \ldots, I_{k-1}$ so that $G_0 + (I_{k_0} + \cdots + I_\ell)$ is $(\ell+1)$-edge-outconnected from r, and $p(I_\ell) \leq 3H(n) \cdot \mathrm{opt}$, $\ell = k_0, \ldots, k-1$. Consequently, $G = G_0 + (I_{k_0} + \cdots + I_{k-1})$ is k-edge-outconnected from r, and

$$p(I_{k_0} + \cdots + I_{k-1}) \leq \sum_{\ell=k_0}^{k-1} p(I_\ell) \leq \sum_{\ell=k_0}^{k-1} 3H(n) \cdot \mathrm{opt} = 3(k-k_0)H(n) \cdot \mathrm{opt}.$$

Following [7], Min-Power $(\ell, \ell+1)$-Edge-Outconnectivity Augmentation is reduced to the problem of finding a min-power cover of an intersecting family. We say that $\emptyset \neq X \subseteq V - r$ is *tight* in G_0 if $d^{in}(X) = \ell$. From Menger's Theorem we have:

Fact 4.1. *Let $G_0 = (V, E_0)$ be ℓ-edge-outconnected from r. Then $G = G_0 + I$ is $(\ell+1)$-edge-outconnected from r if, and only if, I covers all the tight sets in G_0.*

We now see that our augmentation problem is equivalent to the problem of finding a min-power cover of the family $\mathcal{F}$ of tight sets. It is well known (c.f., [7]) that:

Fact 4.2. *The family of tight sets is intersecting.*

It remains to show that given an instance $G_0 = (V, E_0), \ell, r, \mathcal{I}, c$ of Min-Power $(\ell, \ell+1)$-Edge-Outconnectivity Augmentation, Assumptions 1 and 2 are valid for the family $\mathcal{F}$ of tight sets. It is not hard to verify, using elementary constructions, that the algorithm as in Assumption 1 can be implemented using one max-flow computation, and that the algorithm as in Assumption 2 can be implemented using one min-cost $(\ell+1)$-flow computation. However, since here we know in advance a node r not contained in any member of $\mathcal{F}$, we can achieve a more efficient implementation. For that we show how to implement (i) and (ii) in Lemma 3.5 directly, without using Assumptions 1 and 2, as (i) and (ii) in Lemma 3.5 is all we use in our algorithm. The implementation is similar to the one in Lemma 3.5 but uses a simpler procedure to find the max-cores. As before, we assume $\mathcal{F} \leftarrow \mathcal{F}_I$, meaning that the edge set I accumulated during the algorithm is included in G_0. For (i) we need to show that the families $\mathcal{C}(\mathcal{F})$ and $\mathcal{M}(\mathcal{F})$ can be found in polynomial time; we will show that this can be done using $O(n)$ max-flow computations. For (ii) we will show that finding a min-cost $\mathcal{F}(v, C)$-cover for given $v \in V$ and $C \in \mathcal{C}(\mathcal{F})$ can be done using one min-cost $(\ell+1)$-flow computation.

The min-$\mathcal{F}$-cores can be found using $n-1$ max-flow computations using the following standard procedure; see for example [7]. For every $t \in V - r$, compute a maximum rt-flow in G_0 with unit edge capacities. If its value is ℓ, then in the corresponding residual network compute the set $C_t = \{v \in V : t \text{ is reachable from } v\}$. Then, among the sets C_t computed, output the inclusion minimal ones.

After the min-$\mathcal{F}$-cores are found, to find the max-$\mathcal{F}$-core M_C containing a specific min-core $C \in \mathcal{C}(\mathcal{F})$ do the following. Construct a graph G_0' by adding to G_0 an edge from r to every min-core distinct from C. These added edges do not cover any $\mathcal{F}$-core containing C, but they do cover any other member of $\mathcal{F}$. Thus M_C is the largest node subset of $V - r$ of indegree ℓ in G_0' that contains C (or, which is equivalent, some node $t \in C$). Hence M_C can be found using the following known procedure. Choose $t \in C$ and compute a maximum rt-flow in G_0' with unit edge capacities; the max-flow value is ℓ, by the Max-Flow–Min-Cut Theorem. In the corresponding residual network the set of nodes $\{v \in V : v \text{ is not reachable from } r\}$ is the max-core M_C containing C.

Now we show how to find a min-cost $\mathcal{F}(v, C)$-cover for any $v \in V$ and $C \in \mathcal{C}(\mathcal{F})$. Let G_0' be the graph obtained by adding to G_0 an edge from r to every min-core distinct from C and the edge rv. Note that these added edges do not cover any member of $\mathcal{F}(v, C)$, but they do cover all the other members of $\mathcal{F}$. Thus an edge set is a $\mathcal{F}(v, C)$-cover if, and only if, its addition to G'_0 allows a flow of value $\ell + 1$ from r to any node $t \in C$ (with all edges having unit capacities), by the Max-Flow–Min-Cut Theorem. Now construct a flow network by adding $\mathcal{I}$ to G'_0; the edges in H_0 have cost zero, while the edges in $\mathcal{I}$ keep their original cost. Then compute a min-cost $(\ell + 1)$-flow f from r to some $t \in C$. The edge set $\{e \in \mathcal{I} : f(e) = 1\}$ is the desired $\mathcal{F}(v, C)$-cover.

We now give a $(3H(n) + 1)$-approximation algorithm for Min-Power $(\ell, \ell + 1)$-Edge-Connectivity Augmentation. We apply this algorithm sequentially for $\ell = k_0, \ldots, k - 1$ to produce edge sets $I_{k_0}, \ldots, I_{k-1}$ so that $G_0 + (I_{k_0} + \cdots + I_\ell)$ is $(\ell + 1)$-edge-connected, and $p(I_\ell) \leq (3H(n) + 1) \cdot \mathrm{opt}$, $\ell = k_0, \ldots, k - 1$. Consequently, $G = G_0 + (I_{k_0} + \cdots + I_{k-1})$ is k-edge-connected, and

$$p(I_{k_0} + \cdots + I_{k-1}) \leq \sum_{\ell=k_0}^{k-1} p(I_\ell) \leq \sum_{\ell=k_0}^{k-1} (3H(n) + 1) \cdot \mathrm{opt} = (k - k_0)(3H(n) + 1) \cdot \mathrm{opt}.$$

A $(3H(n)+1)$-approximation algorithm for Min-Power $(\ell, \ell+1)$-Edge-Connectivity Augmentation can be deduced from Corollary 1.5 by establishing a reduction to Set-Family Edge-Cover with crossing $\mathcal{F}$. But even simpler would be to describe the algorithm explicitly. Let us say that a graph is *ℓ-edge-inconnected to r* if its reverse graph is ℓ-edge-outconnected from r. It is well known, and easily follows from Facts 1.4 and 4.2, that if I is an inclusion minimal augmenting edge set that increases the edge-inconnectivity of a given directed graph from ℓ to $\ell + 1$ then $d_I(v) \leq 1$ for all $v \in V$. Thus the power of I equals its cost, by Proposition 1.1. The problem of finding a min-cost augmenting edge set that increases the edge-inconnectivity of a given directed graph from ℓ to $\ell + 1$ can be solved in polynomial time, c.f., [7]. Thus we have:

Proposition 4.3. *A min-power augmenting edge set that increases the edge-inconnectivity of a given directed graph by* 1 *can be computed in polynomial time.*

The algorithm for Min-Power $(\ell, \ell + 1)$-Edge-Connectivity Augmentation is as follows.

1. Let r be a node of G. Compute an edge set I' so that $G_0 + I'$ is $(\ell + 1)$-edge-outconnected from r using the algorithm for Min-Power $(\ell, \ell + 1)$-Edge-Outconnectivity Augmentation.
2. Compute a min-power edge set I'' so that $G_0 + I''$ is $(\ell + 1)$-edge-inconnected to r.
3. Output $I = I' + I''$.

Note that $G = G_0 + I$ is both $(\ell + 1)$-edge-outconnected from r and $(\ell + 1)$-edge-inconnected to r. This implies that G is $(\ell + 1)$-edge connected (c.f., [14]), so I is a feasible solution. To bound its power, let OPT be an optimal solution for Min-Power $(\ell, \ell + 1)$-Edge-Connectivity Augmentation. Since $G_0 + OPT$ is $(\ell + 1)$-edge-outconnected from r we have $p(I') \leq 3H(n) \cdot p(OPT) = 3H(n) \cdot \mathrm{opt}$. Since $G_0 + OPT$ is $(\ell + 1)$-edge-inconnected to r we have $p(I'') \leq p(OPT) \leq \mathrm{opt}$. Consequently,

$$p(I) = p(I' + I'') \leq p(I') + p(I'') \leq 3H(n) \cdot \mathrm{opt} + \mathrm{opt} = (3H(n) + 1) \cdot \mathrm{opt}.$$

The proof of Theorem 1.2 is complete.

Acknowledgement

I thank an anonymous referee for many useful comments.

References

[1] E. Althaus, G. Calinescu, I. Mandoiu, S. Prasad, N. Tchervenski, A. Zelikovsky, Power efficient range assignment for symmetric connectivity in static ad-hoc wireless networks, Wireless Networks 12 (3) (2006) 287–299.
[2] G. Calinescu, S. Kapoor, A. Olshevsky, A. Zelikovsky, Network lifetime and power assignment in ad hoc wireless networks, in: ESA, 2003, pp. 114–126.
[3] G. Calinescu, P.J. Wan, Range assignment for biconnectivity and k-edge connectivity in wireless ad hoc networks, Mobile Networks and Applications 11 (2) (2006) 121–128.
[4] I. Caragiannis, C. Kaklamanis, P. Kanellopoulos, Energy-efficient wireless network design, Theory of Computing Systems 39 (5) (2006) 593–617.
[5] J. Edmonds, Matroid intersection, Annals of Discrete Mathematics (1979) 185–204.
[6] A. Frank, Connectivity and network flows, in: R. Graham, M. Grötschel, L. Lovász (Eds.), Handbook of Combinatorics, Elsevier Science, 1995, pp. 111–177.
[7] A. Frank, Increasing the rooted-connectivity of a digraph by one, Mathematical Programming 84 (3) (1999) 565–576.
[8] H.N. Gabow, A representation for crossing set families with application to submodular flow problems, in: SODA, 1993, pp. 202–211.
[9] M.T. Hajiaghayi, G. Kortsarz, V.S. Mirrokni, Z. Nutov, Power optimization for connectivity problems, Mathematical Programming 110 (1) (2007) 195–208.
[10] X. Jia, D. Kim, S. Makki, P.J. Wan, C.W. Yi, Power assignment for k-connectivity in wireless ad hoc networks, Journal of Combinatorial Optimization 9 (2) (2005) 213–222.
[11] S. Khuller, Approximation algorithms for for finding highly connected subgraphs, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-hard problems, PWS, 1995, pp. 236–265 (Chapter 6).
[12] P.N. Klein, R. Ravi, A nearly best-possible approximation algorithm for node-weighted steiner trees, Journal of Algorithms 19 (1) (1995) 104–115.
[13] G. Kortsarz, V.S. Mirrokni, Z. Nutov, E. Tsanko, Approximation algorithms for minimum power degree and connectivity problems, in: LATIN, 2008, pp. 423–435, Algorithmica (in press).
[14] G. Kortsarz, Z. Nutov, Approximating minimum-cost connectivity problems, in: T.F. Gonzalez (Ed.), Approximation Algorithms and Metaheuristics, Chapman & Hall/CRC, 2007 (Chapter 58).

[15] Y. Lando, Z. Nutov, On minimum power connectivity problems, Journal on Discrete Algorithms 8 (2010) 164–173. Preliminary version in ESA 2007, pp. 87–98.
[16] Z. Nutov, Approximating minimum power covers of intersecting families and directed connectivity problems, in: APPROX, 2006, pp. 236–247.
[17] Z. Nutov, Approximating minimum power k-connectivity, in: ADHOC-NOW, 2008, pp. 86–93.
[18] Z. Nutov, Approximating Steiner networks with node weights, in: LATIN, 2008, pp. 411–422.
[19] R. Raz, S. Safra, A sub-constant error-probability low-degree test and a sub-constant error-probability PCP characterization of NP, in: STOC, 1997, pp. 475–484.