

# Approximating Minimum Power Covers of Intersecting Families and Directed Connectivity Problems

Zeev Nutov

The Open University of Israel

`nutov@openu.ac.il`

## 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. Motivated by applications for wireless networks, we consider fundamental directed connectivity network design problems under the power minimization criteria. 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 finding a minimum power subgraph  $G$  of  $\mathcal{G}$  that satisfies some prescribed property. The **Min-Power  $k$ -Outconnected Subgraph (MP $k$ -OS)** problem requires that  $G$  contains  $k$  pairwise internally-disjoint  $rv$ -paths for all  $v \in V - r$ , for a given "root"  $r \in V$ . The **Min-Power  $k$ -Connected Subgraph (MP $k$ -CS)** problem requires that  $G$  contains  $k$  pairwise internally-disjoint paths between every pair of its nodes. In the edge connectivity variant the paths are required to be only edge disjoint. For  $k = 1$  all these problems are at least as hard as the Set-Cover problem and thus have an  $\Omega(\ln |V|)$  approximation threshold. For  $k = \Theta(n)$  the edge connectivity variants are unlikely to admit a polylogarithmic approximation algorithm [23]. We give an  $O(k \ln |V|)$ -approximation algorithms for these four problems. Our algorithms are based on a much 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  $F$  covers  $\mathcal{F}$  if for every  $X \in \mathcal{F}$  there is an edge in  $F$  entering  $X$ .

## 1 Introduction and preliminaries

### 1.1 Problems considered, motivation, and previous work

A large research effort focused on developing algorithms for finding a "cheap" sub-network (sub-graph) 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; the power typically increases at least quadratically in the transmission range. We consider finding a power assignment to

the nodes of a network such that the resulting communication network satisfies prescribed connectivity properties and the total power is minimized. Node-connectivity is more central here than edge-connectivity, as it models stations crashes. For motivation and applications to wireless networks (which is the same as of their min-cost variant for wired networks), see, e.g., [12, 1, 2, 13, 19, 3].

Let  $G = (V, E)$  be a (possibly undirected) 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  $p(G)$  differs from the ordinary cost  $c(G) = \sum_{e \in E} c(e)$  of  $G$  even for unit costs; for unit costs, if  $G$  is undirected, then  $c(G) = |E|$  and  $p(G) = |V|$ . For example, if  $E$  is a perfect matching on  $V$  then  $p(G) = 2c(G)$ . If  $G$  is a clique then  $p(G)$  is roughly  $c(G)/\sqrt{|E|/2}$ . For directed graphs, the ratio between the power and the cost can be equal to the maximum outdegree of a node in  $G$ , e.g., for stars with unit costs. The following statement shows that these are the extremal cases for general edge costs.

**Proposition 1.1 ([13])**  $c(G)/\sqrt{|E|/2} \leq p(G) \leq 2c(G)$  for any undirected graph  $G = (V, E)$ , and if  $G$  is a forest then  $c(G) \leq p(G) \leq 2c(G)$ . For any directed graph  $G$  holds:  $c(G)/d_{\max}(G) \leq p(G) \leq c(G)$ , where  $d_{\max}(G)$  is the maximum outdegree of a node in  $G$ .

A simple connectivity requirement is when there should be a path from a specified node  $r$  to any other node. In this case, the min-cost variant is just the Min-Cost Directed Tree problem which is solvable in polynomial time, while the Min-Power Directed Tree problem is at least as hard as the Set-Cover problem; combined with the result of [26] this implies an  $\Omega(\ln |V|)$ -approximation threshold for this problem (namely, it cannot be approximated within  $C \ln |V|$  for some universal constant  $0 < C < 1$ , unless  $P=NP$ ).

An important network property is fault-tolerance. A graph is  $k$ -outconnected from  $r$  if it has  $k$  internally disjoint  $rv$ -paths for any  $v \in V$ . When the paths only need to be edge-disjoint, the graph is  $k$ -edge-outconnected from  $r$ . A graph is  $k$ -connected (resp.,  $k$ -edge-connected) if it is  $k$ -outconnected (resp.,  $k$ -edge-outconnected) from every node. We consider the following generalization of the problems from [2], where the case  $k = 1$  was studied:

**Min-Power  $k$ -Outconnected Subgraph (MP $k$ -OS):**

*Instance:* A 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  $k$ -outconnected from  $r$  spanning subgraph  $G$  of  $\mathcal{G}$ .

**Min-Power  $k$ -Connected Subgraph (MP $k$ -CS):**

*Instance:* A graph  $\mathcal{G} = (V, \mathcal{E})$  with edge costs  $\{c(e) : e \in \mathcal{E}\}$  and an integer  $k$ .

*Objective:* Find a min-power  $k$ -connected spanning subgraph  $G$  of  $\mathcal{G}$ .

When  $G$  is required to be  $k$ -edge-outconnected or  $k$ -edge-connected, we get the **Min-Power  $k$ -Edge-Outconnected Subgraph (MP $k$ -EOS)** and the **Min-Power  $k$ -Edge-Connected Sub-**

**graph** (MP $k$ -ECS) problems, respectively (for undirected graphs they are equivalent). The "reverse" problems of MP $k$ -OS and MP $k$ -EOS, when  $G$  should contain  $k$  disjoint  $vr$ -paths for every  $v \in V$  are the **Min-Power  $k$ -Inconnected Subgraph** (MP $k$ -IS) and the **Min-Power  $k$ -Edge-Inconnected Subgraph** (MP $k$ -EIS) problems, respectively.

Min-cost versions of these problems were studied extensively for both directed and undirected graphs, see, e.g., [5, 9, 10, 8, 17, 4, 27, 14, 20, 21], and surveys in [7, 16, 22]. For directed graphs the min-cost versions of MP $k$ -OS and MP $k$ -EOS are polynomially solvable, see [5] and [9], respectively; more efficient algorithms are given in [10, 8]. The min-cost  $k$ -edge connected subgraph problem admits a 2-approximation algorithm for both directed and undirected graphs [17]. For the min-cost  $k$ -(node-)connected subgraph problem the best known approximation ratios are:  $O(\ln^2 k \cdot \min\{\frac{n}{n-k}, \frac{\sqrt{k}}{\ln k}\})$  for both directed and undirected graphs [20], and  $O(\ln k)$  for undirected graphs with  $n \geq 2k^2$  [4], where  $n = |V|$ .

In the min-power case, the best known approximation ratio for undirected MP $k$ -ECS is  $O(\min\{k, \sqrt{n}\})$  [13], and for undirected MP $k$ -CS is  $\alpha + O(\ln n)$  [19], where  $\alpha$  is the best known approximation ratio for the min-cost case. Directed min-power connectivity problems seem harder to approximate, and the methods used in [13, 19] do not seem to work for the directed case. For example, for  $k = 1$  undirected MP $k$ -CS/MP $k$ -ECS admits an easy 2-approximation algorithm by just taking a min-cost spanning tree (the 2-approximation follows from Proposition 1.1), while its directed variant is "Set-Cover hard".

The problems MP $k$ -OS and MP $k$ -CS that we study are closely related to the undirected Node Weighted Steiner Forest problem considered by Klein and Ravi [18]; one difference is that in our problems the "weight" of a node  $v$  is not fixed but depends on the chosen edges leaving  $v$ . The Klein-Ravi algorithm [18] uses the set-cover greedy approach [15]. 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  $n$ th Harmonic number). This was improved by Guha and Khuller [11] to  $(1.35 + \epsilon)H(n)$  using a slight generalizations of spiders. For directed MP $k$ -OS with  $k = 1$ , [2] gave a  $2H(n)$ -approximation algorithm using a similar method. They also gave a  $(2H(n) + 1)$ -approximation algorithm for directed MP $k$ -CS/MP $k$ -ECS with  $k = 1$ .

## 1.2 Results in this paper

Henceforth we consider mainly directed graphs, so, unless stated otherwise, "graph" means "directed graph". Suppose that  $\mathcal{G}$  has a subgraph  $G_0 = (V, E_0)$  of power zero which is  $k_0$ -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  $k$ -outconnected from  $r$ . Formally:

**Min-Power  $(k_0, k)$ -Outconnectivity Augmentation** (MP $(k_0, k)$ -OA):

*Instance:* A graph  $G_0 = (V, E_0)$  which is  $k_0$ -outconnected from  $r$ , an edge set  $\mathcal{I}$  on  $V$  with 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  $k$ -outconnected from  $r$ .

In a similar way we define the augmentation versions of MP $k$ -EOS, MP $k$ -CS, and MP $k$ -ECS, respectively:

**Min-Power  $(k_0, k)$ -Edge-Outconnectivity Augmentation** (MP $(k_0, k)$ -EOA);**Min-Power  $(k_0, k)$ -Connectivity Augmentation** (MP $(k_0, k)$ -CA);**Min-Power  $(k_0, k)$ -Edge-Connectivity Augmentation** (MP $(k_0, k)$ -ECA).

In [2], approximation algorithms are given for  $k_0 = 0$  and  $k = 1$ : a  $2H(n)$ -approximation for the Min-Power Directed Tree problem and a  $(2H(n) + 1)$ -approximation for the Min-Power Strongly Connected Subgraph problem. As was mentioned, each one of these problems generalizes the Set-Cover problem (c.f., [2]), and thus the results in [2] are essentially tight up to a constant factor. For arbitrary  $k_0, k$  we prove:

**Theorem 1.2** *There exist approximation algorithms with approximation ratios:*

- (i)  $3(k - k_0)H(n) = O(k \ln n)$  for directed MP $(k_0, k)$ -OA and MP $(k_0, k)$ -EOA;
- (ii)  $(k - k_0)(3H(n) + 1) = O(k \ln n)$  for directed MP $(k_0, k)$ -ECA;
- (iii)  $3(k - k_0)H(n) + k = O(k \ln n)$  for directed MP $(k_0, k)$ -CA.

*Thus each of one of the problems MP $k$ -OS, MP $k$ -CS, MP $k$ -EOS, MP $k$ -ECS on directed graphs admits an  $O(k \ln n)$ -approximation algorithm.*

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 algorithm does not exist: in [23] it is proved that for  $k = \Theta(n)$  directed MP $k$ -EOS, MP $k$ -EIS, and MP $k$ -ECS, cannot be approximated within  $O(2^{\log^{1-\varepsilon} n})$  for any fixed  $\varepsilon > 0$ , unless  $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$ . Note that these hardness results are valid only for *edge-connectivity* and large values of  $k$ . For example, the edge-connectivity problem MP $k$ -EIS admits a  $k$ -approximation algorithm (see Section 4.2) and, in particular, is in P for  $k = 1$ . Even more surprising is that directed MP $k$ -IS (the node-connectivity variant of MP $k$ -EIS) is in P for any  $k$ , see [23]. In [23] is also proved that *undirected* MP $k$ -OS admits an  $O(\ln n)$ -approximation algorithm for any  $k$ .

Table 1 summarizes the currently best known approximation ratios and thresholds for connectivity problems considered. Note again that *directed* MP $k$ -OS and MP $k$ -IS are *not* equivalent; while directed MP $k$ -OS is set-cover hard, directed MP $k$ -IS is in P.

Theorem 1.2 is just a summary of (some) applications of a general approximation algorithm for

Problem	Edge-Connectivity		Node-Connectivity	
	Undirected	Directed	Undirected	Directed
MPk-IS	$2k - 1/3$ [19] $\Omega(\max\{1/\sqrt{\sigma}, \ln n\})$ [23]	$k$ $\Omega(2^{\log^{1-\varepsilon} n})$ [23]	$\min\{k + 4, O(\ln n)\}$ [23] APX for $k = 1$ [13]	in P [23]
MPk-OS	$2k - 1/3$ [19] $\Omega(\max\{1/\sqrt{\sigma}, \ln n\})$ [23]	$O(k \ln n)$ $\Omega(2^{\log^{1-\varepsilon} n})$ [23]	$\min\{k + 4, O(\ln n)\}$ [23] APX for $k = 1$ [13]	$O(k \ln n)$ $\Omega(\ln n)$ for $k = 1$ [2]
MPk-CS	$2k - 1/3$ [19] $\Omega(\max\{1/\sqrt{\sigma}, \ln n\})$ [23]	$O(k \ln n)$ $\Omega(2^{\log^{1-\varepsilon} n})$ [23]	$O(\alpha + \ln n)$ [19] APX for $k = 1$ [13]	$O(k \ln n)$ $\Omega(\ln n)$ for $k = 1$ [2]

Table 1: Currently best known approximation ratios and thresholds for some min-power connectivity problems. Results without references are proved in this paper.  $\sigma$  is the best ratio for the Densest  $k$ -Subgraph problem; currently  $\sigma$  is roughly  $n^{-1/3}$  [6].  $\alpha$  is the best ratio for the Min-Cost  $k$ -Connected Subgraph problem; currently,  $\alpha = O(\ln k \cdot \min\{\sqrt{k}, \frac{n}{n-k} \ln k\})$  for both directed and undirected graphs [21], and  $\alpha = O(\ln k)$  for undirected graphs with  $n \geq k(2k - 1)$  [4].

finding a min-power (edge-)cover of a certain widely studied type of set-families.

**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 intersecting  $X, Y \in \mathcal{F}$ .
- 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 (MPSFC):**

*Instance:* A set-family  $\mathcal{F}$  on a groundset  $V$ , an edge set  $\mathcal{I}$  on  $V$ , and edge costs  $\{c(e) : e \in \mathcal{I}\}$ .

*Objective:* Find a min-power  $\mathcal{F}$ -cover  $I \subseteq \mathcal{I}$ .

We give a  $3H(|V|)$ -approximation algorithm for MPSFC with intersecting  $\mathcal{F}$ , but its polynomial implementation requires that certain queries related to  $\mathcal{F}$  can be answered in polynomial time. We need some definitions in order to present the assumptions we need.

**Definition 1.2** 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. For  $C \in \mathcal{C}(\mathcal{F})$  let  $M_C \in \mathcal{M}(\mathcal{F})$  be the max- $\mathcal{F}$ -core containing  $C$ .

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. It is easy to see that:

**Fact 1.3** Let  $\mathcal{F}$  be an intersecting family. Then the max- $\mathcal{F}$ -cores are pairwise disjoint, and every  $\mathcal{F}$ -core is contained in a unique max- $\mathcal{F}$ -core and contains a unique min- $\mathcal{F}$ -core. Furthermore, if  $X \in \mathcal{F}$  intersects a min-core  $C \in \mathcal{C}(\mathcal{F})$  then  $C \subseteq X$ .

**Definition 1.3** For  $v \in V$  and  $C \in \mathcal{C}(\mathcal{F})$  let  $\mathcal{F}(v, C)$  denote the family of  $\mathcal{F}$ -cores containing  $C$  and not containing  $v$ . Given an edge set  $I$  on  $V$ , 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 well known that if  $\mathcal{F}$  is intersecting, so is  $\mathcal{F}_I$ , for any  $I$ . For any edge set  $I$  on  $V$ , make the following two assumptions (we will show later they are valid for the problems considered, see Section 4.1):

**Assumption 1:**

The families  $\mathcal{C}(\mathcal{F}_I)$  and  $\mathcal{M}(\mathcal{F}_I)$  can be computed in polynomial time.

**Assumption 2:**

Given an edge set  $\mathcal{I}$  on  $V$ , a min-cost  $\mathcal{F}_I(v, C)$ -cover in  $\mathcal{I}$  can be computed in polynomial time for any  $C \in \mathcal{C}(\mathcal{F}_I)$  and  $v \in M_C$ , where  $M_C$  is the max- $\mathcal{F}_I$ -core containing  $C$ .

**Theorem 1.4** MPSFC with intersecting  $\mathcal{F}$  admits a  $3H(|\mathcal{C}(\mathcal{F})|)$ -approximation algorithm under Assumptions 1 and 2.

As shown below, Theorem 1.4 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 nonempty. 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., [8]):

**Fact 1.5** 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 its 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  $\mathcal{F}_r^{out} = \{V - X : X \in \mathcal{F} - \mathcal{F}_r^+\}$ . 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 1.5, we get:

**Corollary 1.6** 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 *intersecting supermodular* if  $f(X) + f(Y) \leq f(X \cap Y) + f(X \cup Y)$  for any intersecting  $X, Y \subset V$ . An edge set  $I$  covers  $f$  if in the graph  $(V, I)$  the indegree of every  $X \subset V$  is at least  $f(X)$ . A  $\{0, 1\}$ -valued set-function is intersecting supermodular if, and only if, its support is an intersecting family. A natural question is whether Theorem 1.4 extends to intersecting supermodular set-functions. As MPk-EOS is a particular case of the problem of finding a min-power cover of an intersecting supermodular set-function, such an extension is unlikely due to the hardness result of [23].

The main tool used to prove Theorem 1.4 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 approach of [18] that uses "set-cover approximation techniques" based on "density" considerations. However, "star-covers" are much more complicated 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 properly decomposed into spiders, see Section 2. Specifically, we use, among others, the method of "uncrossing" sets (c.f., [27]). Unlike [18], and other papers that used the approach of [18], 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. Furthermore, the parts of our decomposition are not node disjoint, but only edge disjoint. 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 [18, 2], where minimal feasible solutions are trees. (However, with some additional effort, it is possible to improve the ratio for MPSFC with intersecting  $\mathcal{F}$  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 [25] 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 [18].

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

A preliminary version of this paper is [24].

### 1.3 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  $X$ ,  $d_E(X) = |\delta_E(X)|$ ,  $\delta_E^{in}(X) = \delta_E(V - X, X)$  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  $v$  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  $I$ . Throughout the paper, let  $\mathcal{G} = (V, \mathcal{E})$  denote the input graph with nonnegative costs on the edges. Let  $n = |V|$  and  $m = |\mathcal{E}|$ . Given  $\mathcal{G}$ , our goal is to find a minimum power spanning subgraph  $G = (V, E)$  of  $\mathcal{G}$  that satisfies some prescribed property. We assume that a feasible solution exists; otherwise our algorithms can be easily modified to return an error message. Let  $\mathbf{opt}$  denote the optimal solution value of an instance at hand.

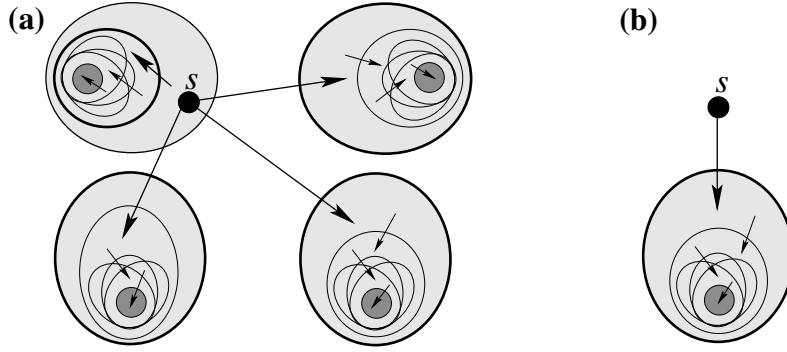


Figure 1: Star-covers (min-cores are shown by dark gray circles).

## 2 Decomposition of covers of intersecting families

We start by briefly describing the decomposition of [18] of a (directed) tree into spiders.

**Definition 2.1** A spider is a directed tree with at most one node of outdegree  $\geq 2$ . A spider decomposition of a directed tree  $T$  is a collection of node disjoint spiders, each of them is a subtree of  $T$ .

**Lemma 2.1** ([18]) Any directed tree  $T$  rooted at  $r$  admits a spider decomposition  $\mathcal{D}$  so that:

- (i)  $\mathcal{D}$  has at most one spider with one leaf (and root  $r$ );
- (ii) every leaf of  $T$  belongs to exactly one spider of  $\mathcal{D}$ .

**Proof:** By induction on the number  $\ell$  of leaves in  $T$ . If  $\ell = 1$  the statement is trivial; then  $T$  is a path with one leaf and root  $r$ . Otherwise,  $T$  has a node  $s$  of outdegree  $\geq 2$  so that the subtree  $S$  that consists of  $s$  and all its descendants is a spider with at least 2 leaves. If  $S$  contains all the leaves of  $T$ , we are done. Otherwise,  $s$  has an ancestor  $s'$  so that the outdegree of  $s'$  is at least 2, but every node in the set  $P$  of the internal nodes on the  $ss'$ -path in  $T$  has outdegree 1. Let  $T' = T - (S \cup P)$ . Note that  $s'$  is *not* a leaf of  $T'$ , hence the sets of leaves of  $T'$  and  $S$  partition the set of leaves of  $T$ . By the induction hypothesis,  $T'$  admits a spider decomposition  $\mathcal{D}'$  as in the Lemma. Thus  $\mathcal{D}' \cup \{S\}$  is also a spider decomposition as in the Lemma.  $\square$

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

**Definition 2.2** Let  $\mathcal{F}$  be an intersecting set-family on  $V$ . An edge set  $S$  on  $V$  is a star-cover (with center  $s$ ) of  $\mathcal{C} \subseteq \mathcal{C}(\mathcal{F})$  if (see Fig 1)  $S$  can be partitioned into  $\mathcal{F}(s, C)$ -covers  $\{S_C : C \in \mathcal{C}\}$ , such that if  $C = \{C\}$  then  $s \notin M_C$ , and such that for every  $C \in \mathcal{C}$ :

- Edges in  $S_C - \delta_S(s)$  has 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  $M_C$ .

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



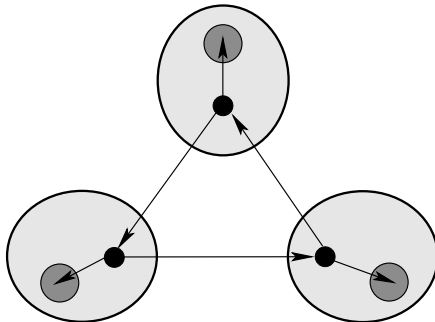


Figure 2: An example showing that the bound  $\lceil 2|\mathcal{C}(\mathcal{F})|/3 \rceil$  in Theorem 2.2 is tight.

**Definition 2.3** Let  $I$  be an  $\mathcal{F}$ -cover of an intersecting set-family  $\mathcal{F}$  on  $V$ . A collection  $\mathcal{S} = \{S_1, \dots, S_q\}$  of star-covers is a star-cover decomposition of  $I$  if  $\text{tails}(S_i) \cap \text{tails}(S_j) = \emptyset$  for all  $i \neq j = 1, \dots, q$ , and if every  $C \in \mathcal{C}(\mathcal{F})$  is covered by at most one member of  $\mathcal{S}$ .

The main result of this section is the following:

**Theorem 2.2 (The Star-Cover Decomposition Theorem)**

Any directed cover  $I$  of an intersecting family  $\mathcal{F}$  admits a star-cover decomposition that 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 2.2 is tight even for laminar families, see Fig.2. In this example, there are three distinct star-covers each covering two min-cores, but there is no star-cover decomposition that covers all min-cores.

The proof of Theorem 2.2 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  $\delta_I^{\text{in}}(W_e) = \{e\}$ ; we call such  $W_e$  a witness set for  $e$ ; note that  $e$  might have several distinct witness sets.

**Lemma 2.3** Let  $W_e, W_f$  be intersecting witness sets of two distinct edges  $e, f \in I$ . Then  $W_e \cap W_f$  is a witness for one of  $e, f$  and  $W_e \cup W_f$  is a witness 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  $W_e, W_f \in \mathcal{F}$  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 sets  $X, Y$  an edge covers one of  $X \cap Y, X \cup Y$  then it also covers 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 2.4**  $d_I^{\text{in}}(C) = 1$  for any min- $\mathcal{F}$ -core  $C$ .

**Proof:** Clearly,  $\delta_I^{\text{in}}(C) \geq 1$  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  $e, f \in \delta_I^{\text{in}}(C)$  for some min- $\mathcal{F}$ -core  $C$ ; let  $W_e, W_f$  be their

witness sets. Then  $C \subseteq W_e, W_f$ , so  $W_e, W_f$  intersect, and  $e, f \in \delta_I^{in}(W_e \cap W_f)$ . This contradicts Lemma 2.3.  $\square$

**Lemma 2.5** *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  $e_1, \dots, e_q$  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  $I_j = \{e_1, \dots, e_j\}$  (and  $I_0 = \emptyset$ ), and  $e_j$  is the unique edge in  $I$  entering  $X_j$ . In particular,  $d_{I(M)}(v) \leq 1$  for every  $v \in M$ .*

**Proof:** Let  $X_1$  be the minimal  $\mathcal{F}$ -core contained in  $M$ . By Corollary 2.4 there is a unique edge in  $I$  entering  $X_1$ , say  $e_1$ . If  $e_1$  covers  $M$ , then  $e_1$  covers all cores contained in  $M$ , by Fact 1.3. Since edges in  $I(M)$  can cover only cores contained in  $M$ , this implies  $I(M) = \emptyset$ , by the minimality of  $I$ ; thus the statement holds in this case. Otherwise, let  $X_2$  be the minimal  $\mathcal{F}_{e_1}$ -core contained in  $M$  and 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  $e_1, \dots, e_q$  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  $I_j = \{e_1, \dots, e_j\}$  and  $e_j$  is the unique edge in  $I$  entering  $X_j$ . The statement follows.  $\square$

**Corollary 2.6** *Let  $M \in \mathcal{M}(\mathcal{F})$ . Then  $p(I(M)) = c(I(M))$ , 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  $I(M) + e_M$  covers  $\{Y \in \mathcal{F} : Y \subseteq M\}$ .*

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 2.2 for simple families. If  $\mathcal{F}$  is not simple, we may replace  $\mathcal{F}$  by the family of  $\mathcal{F}$ -cores; it is easy to see that the latter is intersecting if  $\mathcal{F}$  is.

**Corollary 2.7** *If  $\mathcal{F}$  is simple then  $\delta_I^{in}(M) = \{e_M\}$  for any max- $\mathcal{F}$ -core  $M$ , where  $e_M$  is as in Corollary 2.6.*

**Proof:** Suppose to the contrary that there is  $f \in I - \{e_M\}$  entering  $M$ . Let  $X$  be as in Corollary 2.6. Then  $X$  is a witness set for  $e_M$ . Let  $W_f$  be a witness set for  $f$ . Since  $\mathcal{F}$  is simple,  $W_f \subseteq M$ . But then both  $e_M$  and  $f$  enter  $X \cup W_f$ , contradicting Lemma 2.3.  $\square$

**Remark:** Corollary 2.7 is not true if  $\mathcal{F}$  is not simple. A counterexample is:  $V = \{x, y, u, v\}$ ,  $\mathcal{F} = \{\{x\}, \{x, v\}, \{x, v, y\}\}$ , and  $I = \{uv, xy, yx\}$ . Then  $M = \{x, v\}$  is a max-core,  $e_M = yx$ , but  $uv$  also enters  $M$ . Note that  $\{x, v, y\}$  is a witness set for  $uv$ .

For every  $M \in \mathcal{M}(\mathcal{F})$  let  $e_M$  be the (unique, by Corollary 2.7) edge in  $I$  entering  $M$ . Let us shrink every max-core into a single node, obtaining a graph  $J$ . Since  $\mathcal{F}$  is simple and since  $I$  is a minimal  $\mathcal{F}$ -cover, the edge-set of  $J$  is exactly  $\{e_M : M \in \mathcal{M}\}$ . In  $J$ , the indegree of every node is at most 1, by Corollary 2.7; furthermore, every node of indegree 1 corresponds to a shrunken max-core. Thus  $J$  is a collection of node-disjoint cycles and directed trees. From every cycle of length  $\geq 3$ , delete one edge. For every deleted edge  $e_M$ :

- Remove from  $\mathcal{F}$  all the members of  $\mathcal{F}(M) = \{X \in \mathcal{F} : X \subseteq M\}$ .

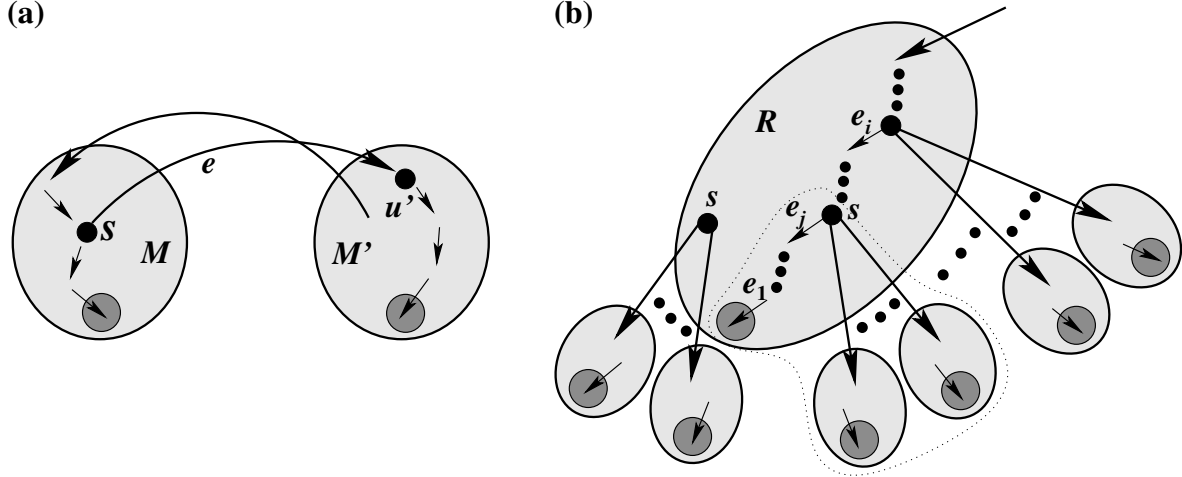


Figure 3: Illustration to the proof of Lemma 2.8. (a)  $J$  is a 2-cycle. (b)  $J$  is a directed tree.

- Remove from  $I$  the set  $I(M)$  of edges with both end-nodes in  $M$ .

Note that at most  $\lfloor |\mathcal{C}(\mathcal{F})|/3 \rfloor$  max-cores were removed in this way, hence at least  $\lceil 2|\mathcal{C}(\mathcal{F})|/3 \rceil$  max-cores remain. It is easy to see that after these changes we obtain a new intersecting family  $\mathcal{F}$ , and a new  $\mathcal{F}$ -cover  $I$ . Hence the following statement implies Theorem 2.2:

**Lemma 2.8** *If  $J$  has no cycles of length  $\geq 3$ , then  $I$  admits a star-cover decomposition that covers all min-cores.*

**Proof:** Since  $J$  consists of node-disjoint 2-cycles and directed trees, it is sufficient to consider two cases: the case when  $J$  is a 2-cycle, and the case when  $J$  is a directed tree.

Suppose that  $J$  is a 2-cycle connecting two max-cores  $M$  and  $M'$  (see Fig. 3(a)). Let  $e = su'$  be the edge of this cycle from  $M$  to  $M'$ , and let  $C$  and  $C'$  be the min-cores contained in  $M$  and  $M'$ , respectively. A star-cover that covers both  $C$  and  $C'$  is obtained by adding to  $\{e\} \cup I(M')$  any  $\mathcal{F}(s, C)$ -cover contained in  $I(M)$ .

Let us now consider the case when  $J$  is a directed tree with root  $r$ . Every node of  $J$  distinct from  $r$  is a shrunken max-core; we thus identify such a node with the corresponding max-core. We prove the statement by induction on the number of max-cores in  $J$ . The induction base is when  $J$  contains a unique max-core  $M$ . Then  $S = I(M) \cup \{e_M\}$  is a star-cover with center  $s = r$ , of the type depicted in Fig. 1(b). Otherwise,  $J$  has a node/max-core  $R$  (a father of the farthest node from  $r$ ) so that all its children are leaves (see Fig. 3(b)). Let  $Y = \text{tails}(\delta(R))$  be the set of tails of the edges in  $J$  leaving  $R$ , and let  $Z = \text{tails}(I(R))$  be the set of tails of edges in  $I(R)$ . Consider two cases:  $Y - Z \neq \emptyset$  and  $Y \subseteq Z$ .

If there is  $s \in Y - Z$ , 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 2.6,  $S$  covers the set  $\mathcal{F}(M) = \{X \in \mathcal{F} : X \subseteq M\}$  for every  $M \in \mathcal{M}(\mathcal{F})$  so that  $e_M \in \delta(s)$ . Now, set

$I' \leftarrow I - S$  and  $\mathcal{F}' \leftarrow \mathcal{F} - \bigcup_{e_M \in \delta(s)} \mathcal{F}(M)$ . 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  $e_1, \dots, e_q$  be an ordering of the edges in  $I(R)$  as in Lemma 2.5, and let  $j$  be the least index so that  $s = \text{tail}(e_j) \in Y$  (see Fig 3(b)). Let  $C$  be the min-core contained in  $R$ , and let  $S_C = \{e_j, \dots, e_1\}$ . By Lemma 2.5,  $S_C$  is an  $(s, C)$ -cover. We now proceed as in the case  $s \in Y - Z$ , except that we set

$$S = \bigcup \{ \{e_M\} \cup I(M) : e_M \in \delta(s) \} \bigcup S_C$$

so now  $S$  also covers  $C$ . We then set  $I' \leftarrow I - (S \cup I(R))$  and  $\mathcal{F}' \leftarrow \mathcal{F} - \bigcup_{e_M \in \delta(s)} \mathcal{F}(M) \cup \mathcal{F}(R)$ , and apply the induction hypothesis in the same way as in the case  $s \in Y - Z$ .  $\square$

The proof of Theorem 2.2 is now complete.

### 3 Covering intersecting families (Proof of Theorem 1.4)

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  $\nu, p$  on groundset  $\mathcal{I}$  given by an evaluation oracle, so that  $\nu$  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,  $\nu$  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  $\rho$ -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

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

$\sigma_I(S)$  is called the *density* of  $S$ . The following statement is known, c.f., [15, 18].

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

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  $S$  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 use  $\mathcal{F}$  instead of  $\mathcal{F}_I$ , and let  $\nu = \nu(\emptyset)$ . Then we need to show that under Assumptions 1 and 2 one can find in polynomial time an edge set  $S$  so that:

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

**Lemma 3.3** *For a star-cover  $S$  that covers  $d$  min-cores let  $\Delta(S) = d - 1$  if  $d \geq 2$  and  $\Delta(S) = 1$  if  $d = 1$ . Then  $\nu - \nu(S) \geq \Delta(S)$ .*

**Proof:** Clearly, min- $\mathcal{F}$ -cores not covered by  $S$  remain min-cores of  $\mathcal{F}_S$ . Suppose that there exists a min- $\mathcal{F}_S$ -core  $X$  that is not a min- $\mathcal{F}$ -core. Let  $C$  be any min- $\mathcal{F}$ -core contained in  $X$ . Clearly,  $C \notin \mathcal{F}_S$ . Thus  $C$  is covered by  $S$ . This implies that  $S$  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  $C' \neq C$ . 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 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  $s \notin M_C$ . 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  $u \in X - M$ . But then  $uv$  covers  $M_C$ , implying, by the definition of a star cover, that  $u = s$ .  $\square$

**Lemma 3.4** *There exists a star-cover  $S$  for which (2) holds.*

**Proof:** Let  $I$  be an optimal  $\mathcal{F}$ -cover, so  $p(I) = \text{opt}$ . By Theorem 2.2,  $I$  admits a star-cover decomposition  $S_1, \dots, S_t$  that cover at least  $2|\mathcal{C}(\mathcal{F})|/3 = 2\nu/3$  min-cores. The statement now follows by a simple averaging argument. Let  $p_i = p(S_i)$  and let  $\Delta_i = \Delta(S_i)$ . We have  $\sum_{i=1}^t p_i \leq p(I) = \text{opt}$  and  $\sum_{i=1}^t \Delta_i \geq (2/3) \cdot \nu/2 \geq \nu/3$ . Thus

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

Consequently, there must be an index  $i$  so that  $p_i/\Delta_i \leq 3p(I)/\nu$ . Let  $S = S_i$ . Then  $\nu - \nu(S) \geq \Delta_i$ , by Lemma 3.3. Thus we have:

$$\sigma(S) = \frac{p(S)}{\nu - \nu(S)} \leq 3 \cdot \frac{p(I)}{\nu} = 3 \cdot \frac{\text{opt}}{\nu}.$$

□

**Lemma 3.5** *A star-cover  $S$  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 done in polynomial time, by Assumption 1. 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 with both endpoints in  $M_C$ , if such exists, and  $w(v) = \infty$  otherwise; this can be done in polynomial time, by Assumption 2. 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 core  $C \in \mathcal{C}(\mathcal{F})$  add a node  $v_C$  and the edge  $sv_C$ . The weight  $w(v_C)$  of  $v_C$  is defined by:

- If  $s \notin M_C$ , then  $w(v_C) = \min\{w(v) : v \in M_C, sv \in \mathcal{E}, c(sv) \leq p(s)\}$  is the minimum weight of a neighbor in  $M_C$  of  $s$ , if such exists, and  $w(v_C) = \infty$  otherwise.
- If  $s \in M_C$ , then let  $s_0 = v_C$ , and  $w(s_0)$  is the minimum cost of an  $\mathcal{F}(s, C)$ -cover, if such exists, and  $w(s_0) = \infty$  otherwise.

We now see that our goal is to compute a sub-star  $S$  of  $T$ ,  $S \neq \{ss_0\}$ , 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  $s_0$  by increasing weight, say  $w(v_1) \leq w(v_2) \leq \dots \leq w(v_q)$ . Let  $W_j = \sum_{i=1}^j w(v_i)$ . Assuming that  $ss_0 \notin S$ , let  $\sigma_1 = p(s) + W_1$  and  $\sigma_j = (p(s) + W_j)/(j - 1)$ ,  $j = 2, \dots, q$ . Assuming that  $ss_0 \in S$ , let  $\sigma_j = (p(s) + w(s_0) + W_j)/j$ ,  $j = 1, \dots, q$ . In both cases, we can find the index  $j$  for which  $\sigma_j$  is minimum, which determines the required star-cover. □

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

**Remark:** The approximation ratio in Theorem 1.4 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 2.2 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  $\ell$ ; hence  $\sigma(Q) \leq p(Q)/(\ell - 1)$  for such  $Q$ . For any  $\ell$ , we can find an ℓ-cycle-cover  $Q$  minimizing  $\sigma(Q)$  in time  $n^{q(\ell)}$ , where  $q(\ell)$  is polynomial in  $\ell$  (details omitted). Setting  $\ell = \lfloor 1/\varepsilon \rfloor$ , we obtain a  $(2 + \varepsilon)H(n)$ -approximation scheme.

## 4 Proof of Theorem 1.2

### 4.1 Part (i)

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

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

A graph  $G = (V, E)$  is  $\ell$ -edge-outconnected from  $r$  to  $T$  if it has  $\ell$  pairwise edge-disjoint  $rt$ -paths for every  $t \in T$ . Using Theorem 1.4, we give a  $3H(n)$ -approximation algorithm for the following problem, that includes both  $\text{MP}(\ell, \ell + 1)\text{-OA}$  and  $\text{MP}(\ell, \ell + 1)\text{-EOA}$ .

*Instance:* A graph  $G_0 = (V, E_0)$  which is  $\ell$ -edge-outconnected from  $r$  to  $T$  and an edge set  $\mathcal{I}$  on  $V$  with costs  $\{c_e : e \in \mathcal{I}\}$  so that every edge in  $\mathcal{I}$  has its head in  $T$ .

*Objective:* Find a min-power edge-set  $I \subseteq \mathcal{I}$  so that  $G = G_0 + I$  is  $(\ell + 1)$ -edge-outconnected from  $r$  to  $T$ .

$\text{MP}(\ell, \ell + 1)\text{-EOA}$  is a special case of this problem when  $T = V$ . For  $\text{MP}(\ell, \ell + 1)\text{-OA}$  apply the following approximation ratio preserving reduction. Given an instance  $G_0 = (V, E_0), \ell, r, \mathcal{I}, c$  of  $\text{MP}(\ell, \ell + 1)\text{-OA}$  obtain an instance  $G'_0 = (V', E'_0), T', \ell', r', \mathcal{I}', c'$  of the above problem as follows. Replace every node  $v \in V$  by the two nodes  $v_t, v_h$  connected by the edge  $v_t v_h$  of cost zero, and replace every edge  $uv \in E_0 \cup \mathcal{I}$  by the edge  $u_h v_t$  having the same cost as  $uv$  (which is zero if  $uv \in E_0$ ). Let  $r' = r_h, T' = \{v_t : v \in V\}$ , and

$$E'_0 = \{u_h v_t : uv \in E_0\} + \{v_t v_h : v \in V\}, \quad \mathcal{I}' = \{u_h v_t : uv \in \mathcal{I}\} .$$

Note that  $c'(e) = 0$  for every  $e \in E'_0$ . This establishes a bijective correspondence between edges in  $\mathcal{I}$  and the edges in  $\mathcal{I}'$ . It is not hard to verify (see [8] for details) that  $G'_0 = (V', E'_0)$  is  $\ell$ -edge-connected from  $r'$  to  $T'$ . Furthermore, if  $I' \subseteq \mathcal{I}'$  corresponds to  $I \subseteq \mathcal{I}$  then:

- (i)  $I$  is a feasible solution if, and only if,  $I'$  is a feasible solution.
- (ii)  $d_I(v) = d_{I'}(v_h)$  and  $d_{I'}(v_t) = 0$  for every  $v \in V$ ; thus  $p(I) = p(I')$ .

We now show that above problem can be reduced to the min-power intersecting family cover problem, so that Assumptions 1 and 2 are valid. We say that  $X \subseteq V - r$  is *tight* in  $G_0$  if  $X \cap T \neq \emptyset$  and  $d^{in}(X) = \ell$ . From Menger's Theorem we have:

**Fact 4.1** *Let  $G_0 = (V, E_0)$  be  $\ell$ -edge-outconnected from  $r$  to  $T$ . Then  $G = G_0 + I$  is  $(\ell + 1)$ -edge-outconnected from  $r$  to  $T$  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 of tight sets. However, since only edges with head in  $T$  can be added, this is equivalent to covering the family:

$$\mathcal{F} = \{X \cap T : X \text{ is tight in } G_0\} . \quad (3)$$

It is well known (c.f. [8]) that:

**Fact 4.2** *The family  $\mathcal{F}$  defined in (3) is intersecting.*

It remains to show that given  $G_0 = (V, E_0), T, \ell, r, \mathcal{I}, c$ , Assumptions 1 and 2 are valid for  $\mathcal{F}$  defined by (3). For Assumption 1 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  $O(|T|)$  max-flow computations. For Assumption 2 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  $|T|$  max-flow computations as follows. For every  $t \in T$ , compute a maximum  $rt$ -flow. If its value is  $\ell$ , then in the corresponding residual network compute the set  $C_t = \{v \in T : t \text{ is reachable from } v\}$ . Then, among the sets  $C_t$  computed, output the inclusion minimal ones.

After the min- $\mathcal{F}$ -cores are found, to find the max- $\mathcal{F}$ -cores, for every 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  $C$ . It is easy to see that these added edges do not cover any  $\mathcal{F}$ -core containing  $C$ , but they cover any other member of  $\mathcal{F}$ . Thus  $M_C$  is the largest node subset of  $V$  of indegree  $\ell$  in  $G'_0$ . Hence  $M_C$  can be computed using the following known procedure. Choose  $t \in C$  and compute a maximum  $rt$ -flow (the max-flow value is  $\ell$ , by the Max-Flow-Min-Cut Theorem); in the corresponding residual network the set of nodes  $\{v \in T : r \text{ is reachable from } v\}$  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  $C \in \mathcal{C}(\mathcal{F})$ . The construction is similar to the previous one: construct a network  $H = G_0 + \mathcal{I}$ , assigning zero costs to edges in  $E_0$ . Then add an edge from  $r$  to every min-core distinct from  $C$ , and 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.

## 4.2 Part (ii)

We give a  $(3H(n) + 1)$ -approximation algorithm for  $\text{MP}(\ell, \ell + 1)$ -ECA. We apply this algorithm sequentially for  $\ell = k_0, \dots, k - 1$  to produce edge sets  $I_{k_0}, \dots, I_{k-1}$  so that  $G_0 + (I_{k_0} + \dots + I_\ell)$  is  $(\ell + 1)$ -edge-connected, and  $p(I_\ell) \leq (3H(n) + 1) \cdot \text{opt}$ ,  $\ell = k_0, \dots, k - 1$ . Consequently,  $G =$



$G_0 + (I_{k_0} + \dots + I_{k-1})$  is  $k$ -edge-connected, and

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

Let us say that a graph is  $\ell$ -edge-inconnected to  $r$  (resp.,  $\ell$ -inconnected to  $r$ ) if its reverse graph is  $\ell$ -edge-outconnected from  $r$  (resp.,  $\ell$ -outconnected from  $r$ ). The problem of finding a min-cost augmenting edge set that increases the inconnectivity (or edge-inconnectivity) of a given directed graph from  $\ell$  to  $\ell + 1$  can be solved in polynomial time, c.f., [8]. Using Fact 1.5 and methods as in the previous section, one can easily deduce:

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

Now a  $(3H(n) + 1)$ -approximation algorithm for  $\text{MP}(\ell, \ell + 1)$ -ECA can be deduced from Corollary 1.6, and explicitly is as follows. Let  $r$  be an arbitrary node of  $G$ .

1. Using the algorithm as in part (i) of Theorem 1.2 compute an edge set  $I'$  so that  $G_0 + I'$  is  $(\ell + 1)$ -edge-outconnected from  $r$ .
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, so  $I$  is a feasible solution. To bound its power, let  $OPT$  be an optimal solution for  $\text{MP}(\ell, \ell + 1)$ -ECA. Since  $G_0 + OPT$  is  $(\ell + 1)$ -edge-outconnected from  $r$  we have  $p(I') \leq 3H(n) \cdot p(OPT) = 3H(n) \cdot \text{opt}$ . Since  $G_0 + OPT$  is  $(\ell + 1)$ -edge-inconnected to  $r$  we have  $p(I'') \leq p(OPT) \leq \text{opt}$ . Consequently,

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

The proof of part (ii) of Theorem 1.2 is complete.

### 4.3 Part (iii)

We give a  $(3(k - k_0)H(n) + k)$ -approximation algorithm for  $\text{MP}(k_0, k)$ -CA. The algorithm is a modification of the  $(k + 1)$ -approximation algorithm of [20] for the Min-Cost  $k$ -Connected Subgraph problem, and is as follows. Let  $S \subseteq V$  be a subset of  $k$  nodes, so  $|S| = k$ . Recall that in [23] is given a polynomial time algorithm for  $\text{MP}k$ -IS – the problem of finding a min-power subgraph which is  $k$ -inconnected to a given node  $s$ .

1. Construct a graph  $\mathcal{G}_r$  by adding to  $\mathcal{G}$  a new node  $r$ , and edges  $\{rs : s \in S\}$  of cost 0. Using the algorithm as in Theorem 1.2 (i) compute an augmenting edge set  $I_r$  so that  $G_r + I_r$  is  $k$ -outconnected from  $r$ , and delete from  $I_r$  the edges leaving  $r$ .
2. For every  $s \in S$  compute an optimal min-power augmenting edge set  $I_s$  so that  $G_0 + I_s$  is  $k$ -inconnected to  $s$  using the algorithm of [23].
3. Output  $I_r + \bigcup_{s \in S} I_s$ .

The fact that the algorithm computes a feasible solution was proved in [20] (this fact is independent from the cost/power of the edge sets computed). For every  $s \in S$  we have  $p(I_s) \leq \text{opt}$ , by [23]. We also have  $p(I_r) \leq 3(k - k_0)H(n) \cdot \text{opt}$ . Consequently,

$$p\left(I_r + \bigcup_{s \in S} I_s\right) \leq (3(k - k_0)H(n) + |S|) \cdot \text{opt} = (3(k - k_0)H(n) + k) \cdot \text{opt} = O(k \ln n).$$

The proof of part (iii) of Theorem 1.2 is complete.

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

## References

- [1] E. Althaus, G. Calinescu, I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment for symmetric connectivity in static ad-hoc wireless networks. *Wireless Networks*, 12(3):287–299, 2006.
- [2] G. Calinescu, S. Kapoor, A. Olshevsky, and A. Zelikovsky. Network lifetime and power assignment in ad hoc wireless networks. In Proc. *European Symposium on Algorithms (ESA)*, LNCS 2832, pages 114–126, 2003.
- [3] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos. Energy-efficient wireless network design. *Theory of Computing Systems*, 39(5):593–617, 2006.
- [4] J. Cheriyan, S. Vempala, and A. Vetta. An approximation algorithm for the minimum-cost  $k$ -vertex connected subgraph. *SIAM Journal on Computing*, 32(4):1050–1055, 2003.
- [5] J. Edmonds. Matroid intersection. *Annals of discrete Math.*, pages 185–204, 1979.
- [6] U. Feige, G. Kortsarz, and D. Peleg. The dense  $k$ -subgraph problem. *Algorithmica*, 29(3):410–421, 2001.
- [7] A. Frank. *Connectivity and network flows*, in *Handbook of Combinatorics*, eds. R. Graham, M. Grötschel, and L. Lovász, pages 111–177. Elsevier Science, 1995.

- [8] A. Frank. Increasing the rooted-connectivity of a digraph by one. *Mathematical Programming*, 84(3):565–576, 1999.
- [9] A. Frank and E. Tardos. An application of submodular flows. *Linear Algebra and its Applications*, 114/115:329–348, 1989.
- [10] H. N. Gabow. A representation for crossing set families with application to submodular flow problems. In Proc. *Symposium on Discrete Algorithms (SODA)*, pages 202–211, 1993.
- [11] S. Guha and S. Khuller. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Inf. Comput.*, 150(1):57–74, 1999.
- [12] M. T. Hajiaghayi, N. Immorlica, and V. S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In Proc. *Mobile Computing and Networking (MOBICOM)*, pages 300–312, 2003.
- [13] M. T. Hajiaghayi, G. Kortsarz, V. S. Mirrokni, and Z. Nutov. Power optimization for connectivity problems. *Math. Programming*, 110(1):195–208, 2007.
- [14] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.
- [15] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computing and System Sciences*, 9(3):256–278, 1974.
- [16] S. Khuller. *Approximation algorithms for finding highly connected subgraphs*, Chapter 6 in *Approximation Algorithms for NP-hard problems*, D. S. Hochbaum Ed., pages 236-265. PWS, 1995.
- [17] S. Khuller and U. Vishkin. Biconnectivity approximations and graph carvings. *Journal of the Association for Computing Machinery*, 41(2):214– 235, 1994.
- [18] C. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *Journal of Algorithms*, 19(1):104–115, 1995.
- [19] G. Kortsarz, V. S. Mirrokni, Z. Nutov, and E. Tsanko. Approximation algorithms for minimum power degree and connectivity problems. Manuscript, 2006.
- [20] G. Kortsarz and Z. Nutov. Approximating node-connectivity problems via set covers. *Algorithmica*, 37:75–92, 2003.
- [21] G. Kortsarz and Z. Nutov. Approximating  $k$ -node connected subgraphs via critical graphs. *SIAM J. on Computing*, 35(1):247–257, 2005.

- [22] G. Kortsarz and Z. Nutov. Approximating minimum cost connectivity problems, in *Approximation Algorithms and Metaheuristics*, T. F. Gonzalez ed., Ch. 58. Chapman & Hall/CRC, 2007.
- [23] Y. Lando and Z. Nutov. On minimum power connectivity problems. To appear in *ESA 2007*.
- [24] Z. Nutov. Approximating minimum power covers of intersecting families and directed connectivity problems. In Proc. *Workshop on Approximation algorithms (APPROX)*, LNCS 4110, pages 236–247, 2006.
- [25] Z. Nutov. Approximating Steiner networks with node weights. *manuscript*, 2007.
- [26] R. Raz and S. Safra. A sub-constant error-probability low-degree test and a sub-constant error-probability PCP characterization of NP. In Proc. *Symposium on the Theory of Computing (STOC)*, pages 475–484, 1997.
- [27] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A primal-dual approximation algorithm for generalized steiner network problems. *Combinatorica*, 15:435–454, 1995.