Activation Network Design Problems

Zeev Nutov

The Open University of Israel

E-mail: nutov@openu.ac.il

3.1 Introduction

In Network Design problems the goal is to select a "cheap" graph that satisfies some property $\mathcal{G}$, meaning that the graph belongs to a family $\mathcal{G}$ of subgraphs of a given graph G. Many fundamental properties can be characterized by **degree demands** (existence of a given number of edges incident to a node) or pairwise **connectivity demands** (existence of a given number of disjoint paths between node pairs). Traditionally, "cheap" means that the edges of the input graph $G = (V, E)$ have costs $\mathbf{c} = \{c_e : e \in E\}$, and the cost of a subgraph of G is the sum of the costs of its edges. Some classic examples of "low demands" problems are Edge-Cover, st-Path, Spanning Tree, Steiner Tree, Steiner Forest, Out-Arborescence, and others. Examples of "high demands" problems are Edge-Multi-Cover, k Disjoint Paths, k-Out-Connected Subgraph, k-Connected Subgraph, and others. See, for example, [34, 13] for polynomial time solvable problems of this type. Here we discuss Activation Network Design problems, where we seek an assignment $\mathbf{a} = \{a_v \geq 0 : v \in V\}$ to the nodes, such that the activated graph $G_{\mathbf{a}} = (V, E_{\mathbf{a}})$ satisfies a given property, and the value $\mathbf{a}(V) = \sum_{v \in V} a_v$ of the assignment is minimized. We now give three examples of such problems.

Node-Weighted Network Design. Here we have node-weights $\mathbf{w} = \{w_v : v \in V\}$ instead of edge-costs. The goal is to find a node subset $V' \subseteq V$ of minimum total weight $\mathbf{w}(V') = \sum_{v \in V'} w_v$, such that the graph (V, E') satisfies the given property, where E' is the set of edges with both endnodes in V'. This can be formulated as an activation problem, where the graph $G_{\mathbf{a}} = (V, E_{\mathbf{a}})$ activated by an assignment $\mathbf{a}$ has edge set $E_{\mathbf{a}} = \{uv : a_u \geq w_u, a_v \geq w_v\}$.

Min-Power Network Design. Consider the following scenario with motivation in wireless networks. We are given a set V of nodes (transmitters) and power thresholds $\mathbf{p} = \{p_{uv} : uv \in V\}$, where p_{uv} is the minimum power (energy level) needed at u to reach v. If u can reach v then we can include the directed edge uv in the activated communication graph. The goal is to find an assignment $\mathbf{a} = \{a_v : v \in V\}$ of power levels to the nodes such that the activated directed graph $G_{\mathbf{a}} = (V, E_{\mathbf{a}})$, where $E_{\mathbf{a}} = \{uv : a_u \geq p_{uv},\ u, v \in V\}$, satisfies the given property. Often one is interested in the undirected network where we have an edge between u and v if each of u, v can reach the other; namely, we have $p_{uv} = p_{vu}$ for all $u, v \in V$, and the activated graph is undirected and has edge set $E_{\mathbf{a}} = \{uv : a_u, a_v \geq p_{uv},\ u, v \in V\}$.

Installation Network Design. Suppose that the installation cost of a wireless network is dominated by the cost of building towers at the nodes for mounting antennas, which in turn is proportional to the height of the towers. An edge uv is activated if the towers at its endpoints u and v are tall enough to overcome obstructions in the middle and establish line-of-sight between the antennas mounted on the towers. This is modeled as each edge uv has a height-threshold requirement h_{uv}, and an edge uv is activated if the scaled heights $s_{uv}a_u, s_{vu}a_v$ at its endpoints sum to at least h_{uv}. Namely, the activated graph $G_{\mathbf{a}} = (V, E_{\mathbf{a}})$ is undirected and has edge set $E_{\mathbf{a}} = \{uv : s_{uv}a_u + s_{vu}a_v \geq h_{uv}\}$.

Panigrahi [33] suggested the following common generalization of these and several other problems.

Definition 3.1 (Panigrahi [33]) *Let $G = (V, E)$ be a graph such that each edge $e = uv \in E$ has an* **activating function** *$f^e = f^{uv}$ from $W^{uv} \subseteq \mathbb{R}^2_+$ to $\{0, 1\}$, where $f^{uv}(x_u, x_v) = f^{vu}(x_v, x_u)$ if e is an undirected edge. Given a non-negative* **assignment** *$\mathbf{a} = \{a_v : v \in V\}$ on V, we say that an edge $uv \in E$ is* **activated** *by $\mathbf{a}$ if $f^{uv}(a_u, a_v) = 1$. Let $E_{\mathbf{a}} = \{uv \in E : f^{uv}(a_u, a_v) = 1\}$ denote the set of edges activated by $\mathbf{a}$. The* **value** *of an assignment $\mathbf{a}$ is $\mathbf{a}(V) = \sum_{v \in V} a_v$.*

> Activation Network Design
>
> *Input:* A graph $G = (V, E)$, a family $\mathbf{f} = \{f^{uv}(x_u, x_v) : e = uv \in E\}$ of activating functions from $W^{uv} \subseteq \mathbb{R}^2_+$ to $\{0, 1\}$ each, and a graph property $\mathcal{G}$ (namely, a family $\mathcal{G}$ of subgraphs of G).
>
> *Output:* An assignment $\mathbf{a} = \{a_v \geq 0 : v \in V\}$ of minimum value $\mathbf{a}(V) = \sum_{v \in V} a_v$ such that the graph $G_{\mathbf{a}} = (V, E_{\mathbf{a}})$ activated by $\mathbf{a}$ satisfies $\mathcal{G}$.

We consider degree and connectivity variants of the above problem. In general, the input graph G may have parallel edges with distinct activation functions. For simplicity of exposition we will assume that G is a simple graph, and use uv to denote the edge from u to v; note that if G is an undirected graph then $uv = vu$ and $f^{uv} = f^{vu}$. In what follows, we will make the following assumptions about the activating functions.

Assumption 1 (Monotonicity)

For every $uv \in E$, f^{uv} is monotone non-decreasing, namely, $f^{uv}(x_u, x_v) = 1$ implies $f^{uv}(y_u, y_v) = 1$ whenever $y_u, y_v \in W^{uv}$, $y_u \geq x_u$, and $y_v \geq x_v$.

Assumption 2 (Polynomial Domain)

For every $uv \in E$, $W^{uv} = W^u \times W^v$ where $|W^u|, |W^v|$ are bounded by a polynomial in $n = |V|$.

For a node v we call W^v the set of **levels of** v. Note that the Monotonicity Assumption holds for the three examples above, and we are not aware of any practical problem when it does not hold. The Polynomial Domain Assumption also holds in many applications; moreover, making this assumption often incurs only a small loss in the approximation ratio. Assumptions 1 and 2 are the default in this survey, but often we can replace the Polynomial Domain Assumption by the weaker assumption:

Assumption 3 (Polynomial Computability)

For any $uv \in E$ we can compute in polynomial time a_u, a_v with $f^{uv}(a_u, a_v) = 1$ and $a_u + a_v$ is minimum.

Let us discuss directed Activation Network Design problems. Then we study the case when each activating function f^{uv} depends only on the assignment at the tail u of the edge uv, so it is a function $f^{uv}(x_u) = f^{uv}(x)$ of one variable. Then by the Monotonicity Assumption each edge uv has a thereshold p_{uv} such that $f^{uv}(x) = 1$ iff $x \geq p_{uv}$. This gives the directed min-power variant discussed earlier, where p_{uv} is the minimum power needed at u to reach to v. Consequently, the directed variant can be stated as follows.

Directed Activation Network Design (Directed Min-Power Network Design)

Input: A graph $G = (V, E)$ with power thresholds $\mathbf{p} = \{p_e : e \in E\}$ and a property $\mathcal{G}$.

Output: An assignment $\mathbf{a} = \{a_v \geq 0 : v \in V\}$ of minimum value $\mathbf{a}(V) = \sum_{v \in V} a_v$ such that the graph $G_{\mathbf{a}} = (V, E_{\mathbf{a}})$ activated by $\mathbf{a}$ satisfies $\mathcal{G}$, where $E_{\mathbf{a}} = \{uv \in E : a_u \geq p_{uv},\ u, v \in V\}$.

We now specify the degree and connectivity problems to be considered. In both types of problems we are given a graph $G = (V, E)$ and certain non-negative integral demands (a.k.a. requirements). In degree problems we have **degree demands** $\mathbf{r} = \{r_v : v \in V\}$ and in connectivity problems we have **connectivity demands** $\mathbf{r} = \{r_{st} : s, t \in V\}$. In both cases we use k to denote the maximum demand. In the case of degree demands we say that a graph (V, J) (or J) **satisfies r** if $\deg_J(v) \geq r_v$ for all $v \in V$, where $\deg_J(v)$ denotes the degree of v in the graph (V, J). In the case of connectivity demands we say that (V, J) (or J) **satisfies r** if the graph (V, J) contains r_{st} pairwise disjoint st-paths for all $s, t \in V$. In edge-connectivity problems the path should be edge disjoint, while in node-connectivity problems the paths should be internally disjoint. In the Edge-Multi-Cover problem we need to satisfy degree demands, while in the Survivable Network problem we need to satisfy connectivity demands. Let us state the min-cost versions of these problems formally.

Min-Cost Edge-Multi-Cover

Input: A graph $G = (V, E)$ with edge-costs $\mathbf{c} = \{c_e : e \in E\}$ and degree demands $\mathbf{r} = \{r_v : v \in V\}$.

Output: A minimum cost edge set $J \subseteq E$ that satisfies $\mathbf{r}$.

Min-Cost Survivable Network

Input: A graph $G = (V, E)$ with edge-costs $\mathbf{c} = \{c_e : e \in E\}$ and connectivity demands $\mathbf{r} = \{r_{st} : s, t \in V\}$.

Output: A minimum cost edge set $J \subseteq E$ that satisfies $\mathbf{r}$.

In activation version of these problems – Activation Edge-Multi-Cover and Activation Survivable Network, instead of edge costs we have activating functions $\mathbf{f} = \{f^e : e \in E\}$ and seek an assignment $\mathbf{a} = \{a_v : v \in V\}$ to the nodes with $\mathbf{a}(V) = \sum_{v \in V} a_v$ minimum such that the graph $G_\mathbf{a} = (V, E_\mathbf{a})$ activated by $\mathbf{a}$ satisfies the demands. In Node-Weighted Edge-Multi-Cover and Node-Weighted Survivable Network we have node-weights $\mathbf{w} = \{w_v : v \in V\}$ and $E_\mathbf{a} = \{uv : a_u \geq w_u, a_v \geq w_v\}$. In Min-Power Edge-Multi-Cover and Min-Power Survivable Network we have power thresholds $\mathbf{p} = \{p_{uv} : uv \in E\}$ and $E_\mathbf{a} = \{uv \in E : a_u, a_v \geq p_{uv}\}$. In the directed general case we also have thresholds $\mathbf{p} = \{p_{uv} : uv \in E\}$ but $E_\mathbf{a} = \{uv \in E : a_u \geq p_{uv}\}$. Also note that in directed Edge-Multi-Cover problems we may have both outdegree and indegree demands $\{(r_v, r^{in}(v)) : v \in V\}$, and $J \subseteq E$ is a feasible solution if $\deg_J(v) \geq r_v$ and $\deg_J^{in}(v) \geq r^{in}(v)$ for all $v \in V$, where $\deg_J(v)$ and $\deg_J^{in}(v)$ denote the outdegree and the indegree of v in the graph (V, J), respectively. In what follows, the Edge-Multi-Cover problem with $0, 1$ demands will be called the Edge-Cover problem.

We summarize the best known ratios for undirected problems with $0, 1$ demands in the following table.

problem/activation fn.	**general**	**node-weighted**	**power**	**cost**
st-Path	in P [33]	in P	in P [2]	in P
Spanning Tree	$O(\ln n)$ [33]	in P	1.5 [17]	in P
Steiner Tree	$O(\ln n)$ [33]	$O(\ln n)$ [21]	$3 \ln 4 - \frac{9}{4} + \epsilon$ [17]	$\ln 4 + \epsilon$ [4]
Steiner Forest	$O(\ln n)$ [33]	$O(\ln n)$ [21]	4 [23]	2 [1]
Edge-Cover	$O(\ln n)$	$O(\ln n)$	1.5 [23]	in P

Table 1.1: Best known approximation ratios for low demands undirected activation problems. The known approximibility of installation problems coincides with those known for the general case. The problems in the table that have ratio $O(\ln n)$ are Set-Cover hard [21], and thus have an approximation threshold $\Omega(\ln n)$.

We now describe the high demands connectivity problems that we consider. Each problem can be defined on directed or undirected graphs. Let us state the node-connectivity versions of these problems.

- k Disjoint Paths: Here $r_{st} = k$ for a given pair of nodes $s, t \in V$ and $r_{uv} = 0$ otherwise, namely, the solution graph should contain k internally disjoint st-paths. For $k = 1$ we get the st-Path problem.
- k-Out-Connectivity and k-In-Connectivity: A graph is **k-out-connected from s** if it contains k internally disjoint paths from s to any other node; similarly, a graph is **k-in-connected to s** if it contains k internally disjoint paths from every node to s (for undirected graphs these two concepts mean the same). In k-Out-Connectivity problems the activated graph should be k-out-connected from s, namely, it should satisfy the node-connectivity demands $r_{st} = k$ for all $t \in V \setminus \{s\}$. In k-In-Connectivity problems the activated graph should be k-in-connected to s. For $k = 1$ we get the Spanning Tree problem in the undirected case, and the problems Out-Arborescence and In-Arborescence in the directed case.
- k-Connectivity: Here the graph should be k-connected, namely it should satisfy the node-connectivity demands $r_{st} = k$ for all $s, t \in V$. For $k = 1$ we get the Spanning Tree problem in the undirected case, and Strong Connectivity problem in the directed case.

The corresponding edge-connectivity problems are k Edge-Disjoint Paths, k-Edge-Out-Connectivity, k-Edge-In-Connectivity, and k-Edge-Connectivity (for undirected graphs the later three problems are equivalent). We abbreviate Edge-Connectivity Survivable Network by EC-Survivable Network. We summarize the best known ratios for high demands undirected and directed activation problems in the following two tables; for the best known ratios for min-cost connectivity problems see surveys [27, 26].

problem/activation fn.	**general**	**node-weighted**	**power**
k Disjoint Paths	2 [31]	2 [31]	2 [18]
k-Out-Connectivity	$O(k \ln n)$ [31]	in P	$\min\{k+1, O(\ln k)\}$ [29, 24, 12]
k-Connectivity	$O(k \ln n)$ [31]	in P	$O\left(\ln k \ln \frac{n}{n-k}\right)$ [12, 32]
k Edge-Disjoint Paths	k [24]	k [24, 30]	k [24]
k-Edge-Connectivity	$O(k \ln n)$ [31]	in P	$\min\{2k-1/2, O(\sqrt{n})\}$ [23, 18]
EC-Survivable Network	$O(k \ln n)$ [31]	$O(k \ln n)$ [30]	$4k$ [23]
Edge-Multi-Cover	$O(k \ln n)$	$O(\ln n)$	$\min\{k+1/2, O(\ln k)\}$ [12]

Table 2.2: Best known ratios for high demands undirected activation problems.

problem		**problem**	**node-connectivity**	**edge-connectivity**
st-Path	in P	k Disjoint Paths	in P [18]	k [24]
In-Arborescence	in P	k-In-Connectivity	in P [24]	k [28]
Out-Arborescence	$O(\ln n)$ [5, 6]	k-Out-Connectivity	$O(k \ln n)$ [31]	$O(k \ln n)$ [28]
Strong Connectivity	$O(\ln n)$ [5, 6]	k-Connectivity	$O(k \ln n)$ [31]	$O(k \ln n)$ [28]

Table 3.3: Best known ratios for directed activation problems.

High demands edge-connectivity problems in the last table are $\Omega\left(2^{\ln^{1-\varepsilon} n}\right)$-hard, assuming NP has no quasi-polynomial time algorithms [18, 24]. The corresponding undirected problems are "Densest k-Subgraph hard" [24, 30], meaning that if the problem admits ratio ρ then the Densest k-Subgraph problem admits ratio $O(\rho^2)$. The Densest k-Subgraph problem was studied extensively, and the best ratio known for it is $\Omega(n^{1/4+\epsilon})$ [3].

Notation. An edge from u to v is denoted by uv. An st**-path** is a path from s to t. For sets A, B of nodes and edges (or graphs) $A \setminus B$ is the set (or graph) obtained by deleting B from A, where deletion of a node implies deletion of all the edges incident to it; similarly, $A \cup B$ is the set (graph) obtained by adding B to A. A set of values given to nodes or edges of a graph is denoted by a bold letter and treated as vector, e.g., $\mathbf{w} = \{w_v : v \in V\}$ usually denotes node-weights, and $\mathbf{w} \cdot \mathbf{x} = \sum_{v \in V} w_v x_v$ for another vector $\mathbf{x} = \{x_v : v \in V\}$. For $V' \subseteq V$ let $\mathbf{w}(V') = \sum_{v \in V'} w_v$. Given a directed/undirected graph or an edge set J, $\delta_J(v)$ is the set of edges in J leaving v and $\deg_J(v) = |\delta_J(v)|$ the **degree** of v in J; $\Delta_J = \max_{v \in V} \deg_J(v)$ denotes the maximum degree of a node w.r.t. J. For directed graphs, degree and out-degree means the same, and $\delta_J^{in}(v)$ and $\deg_J^{in}(v)$ denotes the set of edges in J entering v and the in-degree of v in J, respectively. For a Network Design problem instance k denotes the maximum demand and opt the optimal solution value.

Organization. The rest of this survey is organized as follows. In sections 3.2, 3.3, and 3.4 we give some approximation and exact algorithms using various simple reductions. In Section 3.5 we consider the undirected Min-Power Edge-Multi-Cover problem. In Section 3.6 we survey the algorithm of Klein & Ravi [21] for the Node-Weighted Steiner Forest problem. In Section 3.7 we consider the directed Min-Power k-Edge-Out-Connectivity problem. We conclude in Section 3.8 with some open problems.

3.2 Levels Reduction

Here we discuss a method which we call the **Levels Reduction**, that converts an Activation Network Design problem into a Node-Weighted Network Design problem. This reduction was designed in [24] to solve certain min-power problems, but the reduction and the analysis extend to several activation problems.

Definition 3.2 *The* **levels graph** *of a graph (V, E) with a family $\mathbf{f} = \{f^e : e \in E\}$ of activating functions is a node-weighted graph obtained as follows. For every $v \in V$, take a star with center $v = v(0)$ of weight 0, where for each level $\ell \in W^v \setminus \{0\}$ the star has a leaf $v(\ell)$ of weight ℓ; then replace every edge $uv \in E$ by the edge set $\{u(i)v(j) : f^{uv}(i, j) = 1\}$.*

Note that all nodes of the original graph are present in the levels graph and have weight 0. We consider a natural algorithm that given an Activation Network Design problem computes a solution J' to the corresponding node weighted problem in the levels graph. The algorithm returns the assignment $\mathbf{a}$ defined by the set V' of endnodes of J', namely, $a_v = \max_{v(\ell) \in V'} \ell$ for all $v \in V$, where the maximum taken over the empty set is assumed to be 0. Clearly, we have $\mathbf{a}(V) \leq \mathbf{w}(V')$, but the computed solution may not be feasible. To show that the algorithm is both valid and preserves approximability, we need to show that:

(i) For any feasible solution J' to the node-weighted problem in the levels graph, the assignment $\mathbf{a}$ defined by J' is a feasible solution to the original activation problem.

(ii) For any feasible assignment $\mathbf{a}$ to the original activation problem, the node-weighted problem in the levels graph has a feasible solution of weight at most the value $\mathbf{a}(V)$ of $\mathbf{a}$.

Consider for example the Activation st-Path problem. Let P' be an st-path in the levels graph. If P' contains two leaves of the same star say $v(i)$ and $v(j)$ with $j > i$, then we can "shortcut" P' by replacing the subpath of P' between $v(j)$ and a neighbor q of $v(i)$ on P by the edge $v(j)q$, where q is the successor of $v(i)$ in P if $v(j)$ precedes $v(i)$, and q is a predecessor of $v(i)$ in P otherwise. This is possible since $j > i$, and thus by the Monotonicity Assumption if $v(i)q$ is an edge in the level graph then so is $v(j)q$. Thus we may assume that P' contains at most one leaf from each star. By the construction, the assignment $\mathbf{a}$ defined by J' activates an st-path in the original graph, and property (i) above holds. Conversely, if $\mathbf{a}$ is an assignment activating an st-path P in the original graph, then the edge set $\{(vv(a_v)) : v \in V\} \cup \{(v(a_v)u(a_u)) : uv \in P\}$ in the levels graph contains an st-path of weight $\mathbf{a}(V)$, and property (ii) above holds. Since the Node-Weighted st-Path problem can be solved in polynomial time, we obtain a polynomial time algorithm for the Activation st-Path problem.

Consider the Activation Steiner Forest problem. Recall that in this problem we are given an undirected graph $G = (V, E)$ and a set R of demand node pairs, and seek to activate $J \subseteq E$ such that the graph (V, J) has an st-path for every demand pair $\{s, t\} \in R$. The node-weighted version of the problem admits ratio ratio $2 \ln |U|$, where U is the union of the demand pairs, see Section 3.6. Note that the parameter $|U|$ in the levels graph of G is the same as in G. Thus by the same analysis as in the st-Path case we have:

Theorem 3.1 (Panigrahi [33]) *If* Node-Weighted Steiner Forest *admits ratio* $\rho(|U|)$ *then so is* Activation Steiner Forest, *where* U *is the union of the demand pairs. Thus* Activation Steiner Forest *admits ratio* $2 \ln |U|$.

Let us consider a slightly more complicated example of the Levels Reduction. The Activation k Edge-Disjoint Paths Augmentation problem is a particular case of the Activation k Edge-Disjoint Paths problem when the edge set of the input graph $G = (V, E)$ contains an "initial" edge set $E_0 \subseteq E$ activated by the zero assignment, such that E_0 is a union of $k-1$ pairwise edge-disjoint st-paths. The goal is to find an assignment that activates an edge set $J \subseteq E \setminus E_0$ such that the graph $(V, E_0 \cup J)$ contains k pairwise edge-disjoint st-paths. Clearly, ratio ρ for this augmentation problem implies ratio $k\rho$ for Activation k Edge-Disjoint Paths. Note that for $k = 1$ and $E_0 = \emptyset$ we get the Activation st-Path problem.

Theorem 3.2 (Lando & Nutov [24]) *Undirected* Activation k Edge-Disjoint Paths Augmentation *admits a polynomial time algorithm. Thus undirected* Activation k Edge-Disjoint Paths *admits ratio k.*

Proof: Let D_0 be a set of directed edges obtained by directing $k-1$ pairwise edge-disjoint st-paths in (V, E_0) from t to s. A graph that has both directed and undirected edges will be called a **mixed graph**. In the following algorithm the Activation k Edge-Disjoint Paths Augmentation problem is reduced to the Node-Weighted st-Path problem in a mixed graph. The later problem can be solved in polynomial time by a reduction to the directed Min-Cost st-Path problem (with edge-costs) by elementary constructions: replacing every undirected edge by two opposite directed edges and converting node-weights to edge-costs. Formally, the algorithm is as follows (for illustration see Fig. 3.1).

Algorithm 1: ACTIVATION k EDGE-DISJOINT PATHS AUGMENTATION$((V, E), E_0, \mathbf{f}, \{s, t\}, k)$

1 let G' be the mixed graph obtained from the levels graph of $(V, E \setminus E_0)$ by adding the edges in D_0

2 in the mixed graph G' compute a minimum weight st-path P'

3 return the assignment $\mathbf{a}$ defined by the nodes of P'

We explain why the algorithm is correct. From the correctness of the Ford-Fulkerson algorithm we have: *The graph $(V, E_0 \cup J)$ has k edge-disjoint st-paths if and only if the mixed graph $(V, D_0 \cup J)$ has an st-path.* Thus our problem is equivalent to the Activation st-Path problem in the mixed graph $(V, (E \setminus E_0) \cup D_0)$, where the set D_0 of directed edges is activated by the zero assignment. The Activation st-Path problem in a mixed graph as above is equivalent to the Node-Weighted st-Path problem in the mixed graph G' obtained by adding to the levels graph of $(E \setminus E_0, V)$ the edges in D_0. This follows by the same argument as used for the st-Path problem in ordinary undirected graphs. □

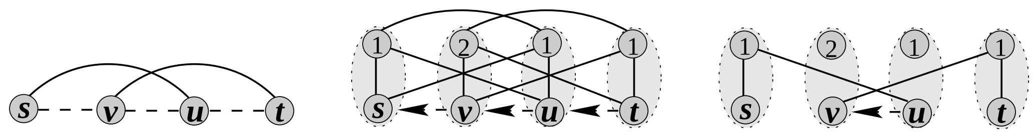

Figure 3.1: Illustration to Algorithm 1 for $k = 2$. Edges in E_0 are shown by dashed lines, v has level set $W^v = \{0, 2\}$ and any other node has level set $\{0, 1\}$. The edge su is activated if $x_s + x_u \geq 1$ and vt is activated if $x_v + x_t \geq 1$. The minimum weight st-path in the levels graph is given by the sequence $s = s(0), s(1), u = u(0), v = v(0), t(1), t = t(0)$, and it defines the assignment $(a_s, a_v, a_u, a_t) = (1, 0, 0, 1)$.

3.3 Min-Cost Reduction

Let $\tau(J) = \tau_{\mathbf{f}}(J)$ denote the optimal value of an assignment activating an edge set J; we use $\tau(e)$ instead of $\tau(\{e\})$. For node-weighted and for min-power problems we will sometimes use the notation $\tau_{\mathbf{w}}(J)$ and $\tau_{\mathbf{p}}(J)$, respectively. For these two problems an optimal assignment $\mathbf{a}$ activating J can be computed in polynomial time; in node weighted problems $a_v = w_v$ if v is an endnode of an edge in J and $a_v = 0$ otherwise, and in min-power problems $a_v = \max\limits_{e \in \delta_J(v)} p_e$. In general activation problems, computing $\tau_{\mathbf{f}}(J)$ is NP-hard. However, we can always find in polynomial time an assignment $\mathbf{a}$ activating J such that $\mathbf{a}(V) \leq \sum\limits_{e \in J} \tau(e)$, as follows. For every $e \in J$ compute an optimal pair $(a_u^e, a_v^e) \in W_u \times W_v$ that activates e, so $f^e(a_u^e, a_v^e) = 1$ and $a_u^e + a_v^e = \tau(e)$; this can be done in polynomial time, by the Polynomial Computability Assumption. The assignment $\mathbf{a}$ defined by $a_v = \max\limits_{e \in \delta_J(v)} a_v^e$ activates J, and has value $\mathbf{a}(V) = \sum\limits_{v \in V} \max\limits_{e \in \delta_J(v)} a_v^e \leq \sum\limits_{e \in J} \tau(e)$.

Assume that we are given an instance of Activation Network Design such that the corresponding min-cost version admits ratio ρ. We will analyze the performance of the following natural algorithm.

Algorithm 2: MIN-COST REDUCTION($G = (V, E), \mathbf{f}, \mathcal{G}$)

1. let $\mathbf{c}$ be edge-costs defined by $c_e = \tau(e)$ for every $e \in E$
2. compute a ρ-approximate $\mathbf{c}$-cost solution $J \in \mathcal{G}$
3. return an assignment $\mathbf{a}$ that activates J of value $\mathbf{a}(V) \leq \sum\limits_{e \in J} \tau(e)$

The algorithm can be implemented in polynomial time, by the Polynomial Computability Assumption. As we shall see, the algorithm has a good performance when inclusionwise minimal feasible solutions have small degree, or are (undirected) forest and the activating functions have small "slope". To state this formally, we need some definitions.

Definition 3.3 *The* **slope** $\theta(e)$ *(of an activating function) of an undirected edge $e = uv$ is defined by* $\theta(e) = \dfrac{\tau(e)}{\min\{\mu_u^e, \mu_v^e\}}$ *where* $\mu_u^e = \min\left\{x_u \in W^u : f(x_u, \max_{w_v \in W^v} w_v) = 1\right\}$ *is the minimum assignment value needed at u to activate e. The slope of $J \subseteq E$ is defined by* $\theta_J = \max_{e \in J} \theta(e)$*, and we denote* $\theta = \theta_E$.

In min-power problems the slope of every edge is exactly 2. In node-weighted problems the slope of an edge $e = uv$ is $\theta(e) = \dfrac{w_u + w_v}{\min\{w_u, w_v\}} = 1 + \dfrac{\max\{w_u, w_v\}}{\min\{w_u, w_v\}}$. For example, if $\dfrac{\max\{w_u, w_v\}}{\min\{w_u, w_v\}} \leq 2$ for every edge $uv \in E$, then the slope of the instance is at most 3. The following statement which particular cases were considered in various papers [31, 7, 18], will enable us to estimate the approximation ratio of Algorithm 2 in terms of various parameters defined.

Lemma 3.1 *Given an instance of* Activation Network Design*, for any $J \subseteq E$ the following holds:*

(i) $\sum_{e \in J} \tau(e) \leq \Delta_J \cdot \tau(J)$ *if J is directed or undirected.*

(ii) $\sum_{e \in J} \tau(e) \leq \theta_J \cdot \tau(J)$ *if J is a forest.*

(iii) $\sum_{e \in J} \tau(e) \leq \theta_J \sqrt{|J|/2} \cdot \tau(J)$ *if J is undirected and has no parallel edges.*

Proof: We prove (i) for the case when J is an an undirected edge set; the proof of the directed case is similar. Let $\mathbf{a}^*$ be an optimal assignment that activates J. Note that $\alpha(e) \leq a_u^* + a_v^*$ for every $e = uv \in J$. This implies

$$\sum_{uv \in J} \tau(uv) \leq \sum_{uv \in J} (a_u^* + a_v^*) = \sum_{v \in V} a_v^* \deg_J(v) \leq \Delta_J \sum_{v \in V} a_v^* = \Delta_J \tau(J) \ .$$

It is sufficient to prove (ii) for the case when J is a tree. Root it at some node s. Then for each $v \neq s$, $\mu(e(v)) \leq a_v^*$ where $e(v)$ is the parent edge of v. This implies

$$\sum_{e \in J} \mu(e) = \sum_{v \in V \setminus \{s\}} \mu(e(v)) \leq \sum_{v \in V} a_v^* = \tau(J) \ .$$

Since $\tau(e) \leq \theta_J \mu(e)$ for every $e \in J$, (ii) follows.

We prove (iii). Let $\mathbf{a}$ be an assignment that activates J. Since $\tau(e) \leq \theta \min\{a_u, a_v\}$ for any edge $e = uv$, it is sufficient to prove that for any undirected simple graph (V, J) with node-weights $\mathbf{w} = \{w_v : v \in V\}$

$$\sum_{uv \in J} \min\{w_u, w_v\} \leq \sqrt{|J|/2} \cdot \mathbf{w}(V) \ .$$

The proof is by induction on the number of distinct weights in $\mathbf{w}$. The induction base is when all nodes have the same weight. Then the inequality above reduces to $|J| \leq n^2/2$, which holds for simple graphs. Otherwise, let q be the difference between the maximum and the second maximum node weight. Let V' be the the set of maximum weight nodes and J' the set of edges in J with both endnodes in V'. Let $\mathbf{w}'$ be defined by $w'_v = w_v - q$ if $v \in V'$ and $w'_v = w_v$ otherwise. By the induction hypothesis we have

$$|J'| \leq \sqrt{|J'|/2} \cdot |V'| \quad \text{and} \sum_{uv \in J} \min\{w'_u, w'_v\} \leq \sqrt{|J|/2} \cdot \mathbf{w}'(V)$$

Applying the induction hypothesis we get

$$\sum_{uv \in J} \min\{w_u, w_v\} = |J'|q + \sum_{uv \in J} \min\{w'_u, w'_v\} \leq \sqrt{|J'|/2} \cdot |V'|q + \sqrt{|J|/2} \cdot \mathbf{w}'(V) \leq \sqrt{|J|/2} \cdot \mathbf{w}(V)$$

as required. □

Corollary 3.1 *For any optimal solution J^*, Algorithm 2 admits the following approximation ratios.*

(i) *For both directed and undirected graphs, ratio $\rho\Delta_{J^*}$.*

(ii) *For undirected graphs, ratio $\rho\theta_{J^*}$ if J^* is a forest.*

(ii) *For undirected simple graphs, ratio $\rho\theta_{J^*}\sqrt{|J^*|/2}$.*

Proof: Note that $c(J) \leq \rho c(J^*)$, since J is a ρ-approximate $\mathbf{c}$-cost solution. Thus we have

$$\mathbf{a}(V) \leq \sum_{e \in J} \tau(e) = c(J) \leq \rho c(J^*) = \rho \sum_{e \in J^*} \tau(e)$$

Now the statement follows by applying Lemma 3.1 on J^*. □

3.3.1 Applications for directed graphs

Here we consider some applications of Corollary 3.1(i) to directed graphs, when inclusionwise minimal feasible solution to the problem at hand have low maximum degree.

Recall that in Theorem 3.2 we considered the undirected Activation k Edge-Disjoint Paths Augmentation problem, when we are given an "initial" edge set $E_0 \subseteq E$ of $k-1$ pairwise edge-disjoint st-paths, and seek to activate an edge set $J \subseteq E \setminus E_0$ such that the graph $(V, E_0 \cup J)$ contains k pairwise edge-disjoint st-paths.

Here we consider the directed variant of this problem. In a similar way we define the Activation k-Edge-In-Connectivity Augmentation problem, where (V, E_0) is $(k-1)$-edge-in-connected to the root s, and $(V, E_0 \cup J)$ should be k-edge-in-connected to s. Note that the case $k = 1$ and $E_0 = \emptyset$ is the Activation In-Arborescence problem. Clearly, ratio ρ for the augmentation version implies ratio $k\rho$ for the "non-augmentation" version.

Corollary 3.2 *The following directed activation problems admit a polynomial time algorithm:* k Edge-Disjoint Paths Augmentation, k-Edge-In-Connectivity Augmentation, *and* k Disjoint Paths.

Proof: For k Edge-Disjoint Paths Augmentation and k-Edge-In-Connectivity Augmentation, it is known that $\Delta_J \leq 1$ holds for any inclusionwise minimal solution J, and the min-cost version of the problem admits a polynomial time algorithm; c.f [13]. Hence we can apply Corollary 3.1(i) with $\Delta = \rho = 1$ and get ratio $\Delta\rho = 1$, namely, a polynomial time algorithm.

Let us consider the k Disjoint Paths problem. We may assume that we know the value a_s^* of some optimal solution $\mathbf{a}^*$; by the Polynomial Domain Assumption there is a polynomial number of choices. We set $a_s = a_s^*$, meaning that the set of edges activated by this assignment are included in any feasible solution, while other edges leaving s are removed. We then apply Algorithm 2 on the modified instance, and now any inclusionwise feasible solution has maximum degree 1. The min-cost version admits a polynomial time algorithm, and thus we get ratio $\rho\Delta = 1$, namely, a polynomial time algorithm. $\square$

In some cases we can achieve a good ratio using Corollary 3.1 after adding a certain set of edges and considering the resulting residual problem.

Theorem 3.3 (Lando & Nutov [24]) *Directed* Activation k-In-Connectivity *admits a polynomial time algorithm.*

Proof: In [24] the following is proved:

Let G' be a directed graph with $\deg_{G'}(v) \geq k$ for all $v \in V \setminus \{s\}$ and let J be an inclusionwise minimal augmenting edge set on V such that $G' \cup J$ is k-inconnected to s. Then $\Delta_J \leq 1$.

The problem of finding a minimum-cost augmenting edge set J as above can be solved in polynomial time, thus by the above result of [24] and Corollary 3.1(i) the activation version of the problem can also be solved in polynomial time. Now we state the algorithm.

Algorithm 3: DIRECTED ACTIVATION k-IN-CONNECTIVITY$(G=(V,E),s,k,\mathbf{f})$

---

1 find a minimum value assignment $\mathbf{a}'$ such that in $(V,E_{\mathbf{a}'})$ every node $v \in V \setminus \{s\}$ has out-degree $\geq k$; namely, a'_v is the k-th least power of an edge in $\delta_E(v)$ for every $v \in V \setminus \{s\}$, and $a'_s = 0$

2 with power thresholds $p'_{uv} = p_{uv} - a'_u$ for all $uv \in E \setminus E_{\mathbf{a}'}$ find an optimal assignment $\mathbf{a}$ that activates an edge set $J \subseteq E \setminus E_{\mathbf{a}'}$ such that $(V, E_{\mathbf{a}'} \cup J)$ is k-inconnected to s

3 return $\mathbf{a}' + \mathbf{a}$

Clearly, the computed assignment is feasible. We explain why the assignment $\mathbf{a}'+\mathbf{a}$ is optimal. Let $\mathbf{a}^*$ be an optimal assignment. Clearly, $\mathbf{a}' \leq \mathbf{a}^*$, hence $\mathbf{a}^* - \mathbf{a}' \geq \mathbf{0}$. Since the assignment $\mathbf{a}$ is optimal, $\mathbf{a}(V) \leq (\mathbf{a}^* - \mathbf{a}')(V)$, hence $(\mathbf{a}+\mathbf{a}')(V) \leq \mathbf{a}^*(V)$, as required. $\square$

3.3.2 Applications for undirected graphs

We consider consequences from Corollary 3.1 for undirected graphs. Among the problems considered in the next Corollary 3.3 is the (undirected) Activation EC-Survivable Network Augmentation problem. In this problem we are given an "initial" graph (V,E_0) and a set of demand node pairs. The goal is to activate an edge set $J \subseteq E \setminus E_0$ such that for every demand pair $\{s,t\}$ the number of pairwise edge-disjoint st-paths in $(V, E_0 \cup J)$ is larger by one than in (V,E_0). Clearly, ratio ρ for this problem implies ratio $k\rho$ for Activation EC-Survivable Network. Also note that for $k=1$ and $E_0=\emptyset$ we get the Activation Steiner Forest problem.

Corollary 3.3 Activation Spanning Tree *admits ratio* θ*,* Activation Steiner Tree *admits ratio* $(\ln 4+\epsilon)\theta$*, and undirected* Activation EC-Survivable Network Augmentation *admits ratio* 2θ*.*

Proof: For each one of the problems, any inclusionwise minimal solution is a forest; for Spanning Tree and Steiner Tree this is obvious, while for EC-Survivable Network Augmentation this is proved in [16]. For the min-cost versions of the problems, the following is known: Spanning Tree admits a polynomial time algorithm, Steiner Tree admits ratio $\ln 4+\epsilon$ [4], and EC-Survivable Network Augmentation admits ratio 2 [16]. For the activation variants we get ratios larger by a factor of θ, by Corollary 3.1(ii). $\square$

In min-power problems $\theta = 2$, and thus Corollary 3.3 implies that Min-Power Spanning Tree admits ratio 2 and Min-Power Steiner Tree admits ratio $2\ln 4+\epsilon$. We have the following improvement over these ratios:

Theorem 3.4 (Grandoni [17]) Min-Power Spanning Tree *admits ratio* 1.5 *and* Min-Power Steiner Tree *admits ratio* $3\ln 4 - \frac{9}{4} + \epsilon < 1.909$.

The proof of Theorem 3.4 relies on a method that is beyond the scope of this survey.

We now continue our list of applications of Corollary 3.1.

Corollary 3.4 *Undirected* Activation k Disjoint Paths *admits ratio* 2.

Proof: We "guess" the values of a_s^* and a_t^* of some optimal solution $\mathbf{a}^*$, as in the proof of Corollary 3.2 for the k Disjoint Paths problem. For every edge $sv \in E$ we set the cost of sv to be $c(sv) = \min\{x_v : f^{sv}(a_s^*, x_v) = 1\}$, and for every edge $ut \in E$ we set $c(ut) = \min\{x_u : f^{ut}(a_u, a_t^*) = 1\}$. Now we apply Algorithm 2 with these costs and with costs $\tau(e)$ for edges that are not incident to s or to t. The min-cost solution J computed can be activated by an assignment $\mathbf{a}$ of value $c(J) + a_s^* + a_t^*$, while by an analysis similar to the proof of Lemma 3.1(i) we get that $a_s^* + a_t^* + c(J) \leq a_s^* + a_t^* + 2\mathbf{a}^*(V \setminus \{s,t\}) \leq 2\mathbf{a}^*(V)$. □

Corollary 3.5 ([18, 24]) *For undirected graphs, if* Activation Edge-Multi-Cover *admits ratio* $\alpha(k)$ *then:*

(i) Activation k-Connectivity *admits ratio* $\alpha(k) + O\left(\theta \ln k \ln \frac{n}{n-k}\right)$ *and ratio* $\alpha(k) + 6\theta$ *if* $n \geq k^3$.

(ii) Activation k-In-Connectivity *admits ratio* $\alpha(k) + 2\theta$.

(iii) Activation k-Edge-Connectivity *admits ratio* $\alpha(k) + O(\theta\sqrt{n})$.

Proof: We compute an $\alpha(k)$-approximate solution I to Activation Edge-Multi-Cover with demands $r_v = k$ for all $v \in V$. Clearly, $\tau(I) \leq \alpha\mathsf{opt}$. Then we use Algorithm 2 to find an inclusionwise minimal augmenting edge set J such that $(V, I \cup J)$ satisfies the connectivity demands.

In the case of k-Connectivity J is a forest [25], the min-cost version admits ratio $O\left(\ln k \ln \frac{n}{n-k}\right)$ [32] and ratio 6 if $n \geq k^3$ [10] (see also [15]). In the case of k-In-Connectivity J is a forest [8] and the min-cost version admits ratio 2 [14]. In both cases we get the stated ratio from Corollary 3.1(ii). In the case of k-Edge-Connectivity J has at most $\frac{kn}{k+1} < n$ edges [9], the min-cost version admit ratio 2 [20], and the statement follows from Corollary 3.1(iii). □

Since Min-Power Edge-Multi-Cover admits ratio $O(\ln k)$ [12], and in min-power problems $\theta = 2$, we get:

Corollary 3.6 *For undirected graphs,* Min-Power k-Connectivity *admits ratio* $O\left(\ln k \ln \frac{n}{n-k}\right)$, Min-Power k-In-Connectivity *admits ratio* $O(\ln k)$*, and* Min-Power k-Edge-Connectivity *admits ratio* $O(\sqrt{n})$.

3.4 Bidirection Reduction

Finally, we discuss factors invoked in the approximation ratio when undirected min-power problems are reduced to directed ones. The **bidirected graph** of an undirected graph (V, J) with edge costs is a directed graph obtained by replacing every undirected edge $e = uv$ of J by two opposite directed edges uv and vu each having the same cost as e. Clearly, if (V, D) is a bi-direction of (V, J), then $\mathbf{a}$ activates J if and only if $\mathbf{a}$ activates D. The **underlying graph** of a directed graph D is obtained from D by ignoring the directions (but keeping costs) of the edges. We will analyze the performance of the following natural algorithm.

Algorithm 4: BIDIRECTION REDUCTION($G = (V, E), \mathbf{p}, \mathcal{G}$)

1 let $\mathcal{D}$ be a family of subgraphs of the bidirected graph of G such that the following holds:
(i) the underlying graph of every $D \in \mathcal{D}$ is in $\mathcal{G}$ (ii) the bidirected graph of every $J \in \mathcal{G}$ is in $\mathcal{D}$

2 compute a ρ-approximate min-power subgraph $D \in \mathcal{D}$

3 return the underlying graph J of D and an assignment $\mathbf{a}$ of value $\tau_{\mathbf{p}}(J)$ activating J

The following statement will enable us to estimate the approximation ratio of Algorithm 4.

Lemma 3.2 ([29]) $\tau_{\mathbf{p}}(J) \leq (\Delta_D + 1)\tau_{\mathbf{p}}(D)$ *if* (V, J) *is the underlying graph of a directed graph* (V, D)*, where* Δ_D *is the maximum out-degree in the graph* (V, D).

Proof: By induction on $|D|$. For $|D| = 1$ the statement is obvious. Otherwise, let $v \in V$ be a node in (V, D) of maximum power p. Let D' be obtained from D by removing the edges leaving v, and let (V, J') be the underlying graph of (V, D'). Then $\tau_{\mathbf{p}}(D) = \tau_{\mathbf{p}}(D') + p$ and $\tau_{\mathbf{p}}(J) \leq \tau_{\mathbf{p}}(J') + (\Delta_D + 1)p$. By the induction hypothesis $\tau_{\mathbf{p}}(J') \leq (\Delta_{D'} + 1)\tau_{\mathbf{p}}(D')$. Thus we get

$$\tau_{\mathbf{p}}(J) \leq \tau_{\mathbf{p}}(J') + (\Delta_D + 1)p \leq (\Delta_{D'} + 1)\tau_{\mathbf{p}}(D') + (\Delta_D + 1)p \leq (\Delta_D + 1)(\tau_{\mathbf{p}}(D') + p) = (\Delta_D + 1)\tau_{\mathbf{p}}(D) \ ,$$

as required. $\square$

Lemma 3.3 ([29]) *Algorithm 4 admits ratio* $\rho(\Delta_D + 1)$.

Proof: Since $D \in \mathcal{D}$ we have $J \in \mathcal{G}$, by property (i) of $\mathcal{D}$; hence the computed solution J is feasible. We prove the approximation ratio. Let J^* be an optimal solution to the undirected instance and let D^* be the bi-direction of J^*. Then $\tau_{\mathbf{p}}(D) \leq \tau_{\mathbf{p}}(D^*)$, by property (ii) of $\mathcal{D}$. Applying Lemma 3.2 we get $\tau_{\mathbf{p}}(J) \leq (\Delta_D + 1)\tau_{\mathbf{p}}(D) \leq (\Delta_D + 1)\tau_{\mathbf{p}}(D^*) = (\Delta_D + 1)\tau_{\mathbf{p}}(J^*)$, as required. $\square$

Corollary 3.7 *Undirected* Min-Power Edge-Multi-Cover *admits ratio* $k + 1$.

Proof: Let $\mathcal{D}$ be the family of subgraphs of the bi-direction of G that are $\mathbf{r}$-edge-covers. Then properties (i) and (ii) hold for $\mathcal{D}$, and $\Delta_D = k$ for every inclusionwise minimal member $D \in \mathcal{D}$. Directed Edge-Multi-Cover admits a polynomial time algorithm, c.f. [34, 13]. Thus we can apply Lemma 3.3 with $\rho = 1$ and $\Delta_D = k$ and get ratio $k + 1$. $\square$

Corollary 3.8 *Undirected* Min-Power k-In-Connectivity *admits ratio* $k + 1$.

Proof: Let $\mathcal{D}$ be the family of subgraphs of the bi-direction of G that are k-in-connected to s. Then properties (i) and (ii) hold for $\mathcal{D}$, and $\Delta_D = k$ for every inclusionwise minimal member $D \in \mathcal{D}$. Thus we can apply Lemma 3.3 with $\rho = 1$ and $\Delta_D = k$ and get ratio $k + 1$. $\square$

3.5 Undirected Min-Power Edge-Multi-Cover problems

Ratio 2 for Min-Power Edge-Cover follows from Corollary 3.1. We survey the following improvement.

Theorem 3.5 (Kortsarz & Nutov [23]) Undirected Min-Power Edge-Cover *admits ratio* $3/2$.

Proof: Given a node subset S we say that an edge set I is an S**-edge-cover** if $\delta_I(v) \neq \emptyset$ for all $v \in S$. Let $S = \{v \in V : r_v = 1\}$ be the set of nodes we need to cover. The algorithm is as follows.

Algorithm 5: MIN-POWER EDGE-COVER$(G = (V, E), \mathbf{p}, S)$ (ratio 3/2)

---
1 for all $u, v \in S$ (possibly $u = v$) compute an optimal $\{u, v\}$-edge-cover J_{uv}

2 let (S, E') be a complete graph with all loops and edge costs $c_{uv} = \tau_{\mathbf{p}}(J_{uv})$ for all $u, v \in S$

3 compute a minimum $\mathbf{c}$-cost S-edge-cover $J' \subseteq E'$

4 return an optimal assignment activating $J = \bigcup_{uv \in J'} J_{uv}$

Step 1 can be implemented in polynomial time since any inclusionwise minimal $\{u,v\}$-edge-cover has at most 2 edges. Other steps of the algorithm can also be implemented in polynomial time. For the approximation ratio we prove that:

For any S-edge-cover $I \subseteq E$ there exist an S-edge-cover $I' \subseteq E'$ such that $\mathbf{c}(I') \leq \frac{3}{2}\tau_{\mathbf{p}}(I)$.

In particular, if I is an optimal edge-cover and J is the edge set computed by the algorithm then we get

$$\tau_{\mathbf{p}}(J) \leq \sum_{uv \in J'} \tau_{\mathbf{p}}(J_{uv}) = \mathbf{c}(J') \leq \mathbf{c}(I') \leq \frac{3}{2}\tau_{\mathbf{p}}(I) \ .$$

It is sufficient to prove existence of I' as above for the case when I is a star with all leaves in S, since any inclusionwise minimal S-edge-cover is a union of such node disjoint stars. Let v_0 be the center of the star. Let $e_1 = v_0v_1, \ldots, e_d = v_0v_d$ be the edges of I sorted by non-increasing powers, so $p_1 \leq p_2 \leq \cdots \leq p_d$, where $p_i = p(v_0v_i)$ for $i = 1, \ldots, d$. Note that $2p_{i-1} + p_i \leq \frac{3}{2}(p_{i-1} + p_i)$ for all i and that $\tau_{\mathbf{p}}(I) = \sum_{i=1}^{d} p_i + p_d$.

If d is odd then we let $I' = \{v_0v_1, v_2v_3, \ldots, v_{d-1}v_d\}$. Then:

$$\mathbf{c}(I') \leq 2p_1 + p_2 + 2p_3 + p_4 + \cdots + 2p_{d-2} + p_{d-1} + 2p_d \leq \frac{3}{2}\sum_{i=1}^{d} p_i + \frac{1}{2}p_d = \frac{3}{2}\tau_{\mathbf{p}}(I) - p_d$$

If d is even then we let $I' = \{v_0v_1, v_0v_2\} \cup \{v_3v_4, \ldots, v_{d-1}v_d\}$. Then:

$$\mathbf{c}(I') \leq 2p_1 + 2p_2 + p_3 + \cdots + 2p_{d-2} + p_{d-1} + 2p_d \leq \frac{3}{2}\sum_{i=1}^{d} p_i + \frac{1}{2}(p_1 + p_d) \leq \frac{3}{2}\tau_{\mathbf{p}}(I) - \frac{1}{2}p_d \ .$$

In both cases $\mathbf{c}(I') \leq \frac{3}{2}\tau_{\mathbf{p}}(I) - \frac{1}{2}p_d \leq \frac{3}{2}\tau_{\mathbf{p}}(I)$, as claimed. □

By a similar method, we have the following generalization:

Theorem 3.6 (Cohen & Nutov [12]) Min-Power Edge-Multi-Cover *admits ratio $k + 1/2$.*

For small values of k, e.g. for $k \leq 6$, the ratio $k + 1/2$ is currently the best known one. Based on an earlier work on the Min-Power Edge-Multi-Cover problem by Hajiaghayi, Kortsarz, Mirrokni, and Nutov [18] that obtained ratio $O(\ln^4 n)$, and Kortsarz, Mirrokni, Nutov, and Tsanko [22] that obtained ratio $O(\ln n)$, the following is proved in [12].

Theorem 3.7 (Cohen & Nutov [12]) Min-Power Edge-Multi-Cover *admits ratio $O(\ln k)$.*

In the rest of this section we survey the proof of Theorem 3.7. We start by a standard reduction. Add a copy V' of V and replace every edge uv by the edges $u'v$ and uv', each of the same power as uv, where

v' denotes the copy of v. It is easy to see that ratio ρ for the obtained instance implies ratio 2ρ for the original instance. We thus obtain a Bipartite Min-Power Edge-Multi-Cover instance, when the input graph is bipartite with sides V and V' and the demands are $\mathbf{r} = \{r_v : v \in V\}$ (nodes in V' have no demands).

Lemma 3.4 *Let J' be an edge set obtained by picking r_v least power edges in $\delta_E(v)$ for every $v \in V$, and let $\mathbf{a}'$ be an optimal assignment activating J'. Then $\mathbf{a}'(V) \leq \mathsf{opt}$ and $\mathbf{a}'(V') \leq \sum_{v \in V} a'_v r_v \leq k \cdot \mathsf{opt}$.*

Proof: It is clear that $\mathbf{a}'(V) \leq \mathsf{opt}$. Also, $\sum_{v \in V} a'_v r_v \leq \mathbf{a}'(V) \cdot \max_{v \in V} r_v \leq k \cdot \mathsf{opt}$. Finally, since no edge joins two nodes in V', we have $\mathbf{a}'(V') \leq \mathbf{p}(J') \leq \sum_{v \in V} a'_v r_v$. $\square$

Given a partial solution J to our problem, let $\mathbf{r}^J = \{r_v^J : v \in V\}$ be the residual demands w.r.t. J, where $r_v^J = \max\{r_v - \deg_J(v), 0\}$. Given node weights $\mathbf{w} = \{w_v : v \in V\}$ we denote by $\mathbf{w} \cdot \mathbf{r}^J = \sum_{v \in V} w_v r_v^J$ the total weighted residual demand, and call $\mathbf{w} \cdot (\mathbf{r} - \mathbf{r}^J)$ the amount of "weighted demand covered" by J. The main step of the algorithm is given in the following lemma.

Lemma 3.5 *There exists a polynomial time algorithm that given a* Bipartite Min-Power Edge-Multi-Cover *instance with node weights $\mathbf{w} = \{w_v : v \in V\}$ and a parameter $\gamma > 1$, returns an edge set $I \subseteq E$ such that $\tau_{\mathbf{p}}(I) \leq (\gamma + 1)\mathsf{opt}$ and $\mathbf{w} \cdot \mathbf{r}^I \leq \alpha(\mathbf{w} \cdot \mathbf{r})$, where $\alpha = 1 - \left(1 - \frac{1}{\gamma}\right)\left(1 - \frac{1}{e}\right)$.*

Proof: We describe an algorithm that given an integer τ returns an assignment $\mathbf{a}$ and $I \subseteq E_{\mathbf{a}}$ such that:

(i) $\mathbf{a}(V) \leq \gamma\tau$ and $\mathbf{a}(V') \leq \tau$.

(ii) If $\tau \geq \mathsf{opt}$ then $\mathbf{w} \cdot \mathbf{r}^I \leq \alpha(\mathbf{w} \cdot \mathbf{r})$, where $\alpha = 1 - \left(1 - \frac{1}{\gamma}\right)\left(1 - \frac{1}{e}\right)$.

With such an algorithm, we use binary search to find the least integer τ for which an edge set I satisfying $\mathbf{w} \cdot \mathbf{r}^I \leq \alpha(\mathbf{w} \cdot \mathbf{r})$ is returned. Then $\tau \leq \mathsf{opt}$, and we have both $\tau_{\mathbf{p}}(I) \leq (\gamma+1)\mathsf{opt}$ and $\mathbf{w} \cdot \mathbf{r}^I \leq \alpha(\mathbf{w} \cdot \mathbf{r})$.

In [35] it is shown that the following problem admits ratio $1 - 1/e$.

Bipartite Power-Budgeted Maximum Edge-Multi-Coverage

Instance: A bipartite graph $G = (V \cup V', F)$ with edge-powers $\{p_e : e \in F\}$, node-weights $\{w_v : v \in V\}$, degree bounds $\{r_v : v \in V\}$, and a budget τ.

Objective: Find $I \subseteq F$ with $\sum_{v \in V'} \max_{e \in \delta_I(v)} p_e \leq \tau$ that maximizes $\sum_{v \in V} w_v \cdot \min\{\deg_I(v), r_v\}$.

The algorithm computes a $\left(1-\frac{1}{e}\right)$-approximate solution $I \subseteq F$ to the above problem with

$$F = \bigcup_{v \in V} \left\{ e \in \delta_E(v) : p_e \leq \frac{w_v r_v}{\mathbf{w} \cdot \mathbf{r}} \cdot \gamma\tau \right\} .$$

Let $\mathbf{a}$ be an optimal assignment activating I, namely $a_v = \max_{e \in \delta_I(v)} p_e$ for every $v \in V \cup V'$. Clearly, $\mathbf{a}(V) \leq \sum_{v \in V} \frac{w_v r_v}{\mathbf{w} \cdot \mathbf{r}} \cdot \gamma\tau = \gamma\tau$ and $\mathbf{a}(V') \leq \tau$.

We prove that if $\tau \geq \mathsf{opt}$ then $\mathbf{w} \cdot \mathbf{r}^I \leq \alpha(\mathbf{w} \cdot \mathbf{r})$. Let J^* be an optimal solution to our Bipartite Min-Power Edge-Multi-Cover instance. Let $B = \{v \in V : \delta_{J^* \setminus F}(v) \neq \emptyset\}$. For any assignment $\mathbf{a}$ activating $J^* \setminus F$ we have $a_v \geq \frac{w_v r_v}{\mathbf{w} \cdot \mathbf{r}} \cdot \gamma\tau$ for all $v \in B$. This implies $\tau \geq \tau_{\mathbf{p}}(J^* \cap F) \geq \sum_{v \in B} \frac{w_v r_v}{\mathbf{w} \cdot \mathbf{r}} \cdot \gamma\tau$, namely, $\sum_{v \in B} w_v r_v \leq \frac{\mathbf{w} \cdot \mathbf{r}}{\gamma}$. Since the amount of weighted demand covered by $J^* \setminus F$ is at most $\sum_{v \in B} w_v r_v$, the amount of weighted demand covered by $J^* \cap F$ is at least $\left(1-\frac{1}{\gamma}\right)(\mathbf{w} \cdot \mathbf{r})$. Since $\tau_{\mathbf{p}}(J^* \cap F) \leq \tau_{\mathbf{p}}(J^*) \leq \tau$, the amount of weighted demand I covers is at least $\left(1-\frac{1}{\gamma}\right)\left(1-\frac{1}{e}\right)(\mathbf{w} \cdot \mathbf{r})$. Consequently, $\mathbf{w} \cdot \mathbf{r}^I \leq \alpha(\mathbf{w} \cdot \mathbf{r})$. $\square$

Theorem 3.7 is deduced from Lemmas 3.4 and 3.5 as follows. We let γ to be a constant strictly greater than 1, say $\gamma = 2$. Then $\alpha = 1 - \frac{1}{2}\left(1-\frac{1}{e}\right)$. For $v \in V$ we set w_v to be the r_v-th least power of an edge in $\delta_E(v)$, and apply the algorithm from Lemma 3.5 iteratively $\lceil \log_{1/\alpha} k \rceil = O(\ln k)$ times. We then extend the partial solution computed to a feasible solution by adding an edge set as in Lemma 3.4.

Algorithm 6: MIN-POWER EDGE-MULTI-COVER$(G = (V \cup V', E), \mathbf{p}, \mathbf{r})$ (ratio $O(\ln k)$)

1 $J \leftarrow \emptyset$

2 for every $v \in V$ set w_v to be the r_v-th least power of an edge in $\delta_E(v)$

3 repeat $\lceil \log_{1/\alpha} k \rceil$ times:

compute an edge set I as in Lemma 3.5 and update: $\mathbf{r} \leftarrow \mathbf{r}^I$, $J \leftarrow J \cup I$, $E \leftarrow E \setminus I$

4 compute an edge set J' as in Lemma 3.4

5 return $J \cup J'$ and an optimal assignment $\mathbf{a}$ activating $J \cup J'$

It is clear that the algorithm computes a feasible solution. We prove the approximation ratio. At each iteration in the loop of step 3 we compute an edge set I with $\tau_{\mathbf{p}}(I) \leq (1+\gamma)\mathsf{opt}$ and add I to J. We apply this $\lceil \log_{1/\theta} k \rceil$ times, hence $\tau_{\mathbf{p}}(J) \leq \lceil \log_{1/\alpha} k \rceil (1+\gamma)\mathsf{opt} = O(\ln k)\mathsf{opt}$. We show that $\tau_{\mathbf{p}}(J') \leq 2\mathsf{opt}$. Let $\mathbf{a}'$ be an optimal assignment activating J'. By Lemma 3.4 $\mathbf{a}'(V) \leq \mathsf{opt}$. We show that $\mathbf{a}'(V') \leq \mathsf{opt}$. Note that $a'_v \leq w_v$ for every $v \in V$. Thus by Lemma 3.4 we have $\mathbf{a}'(V') \leq \sum_{v \in V} a'_v r_v^J \leq \sum_{v \in V} w_v r_v^J = \mathbf{w} \cdot \mathbf{r}^J$. We claim that $\mathbf{w} \cdot \mathbf{r}^J \leq \mathsf{opt}$. By applying Lemma 3.4 on the initial instance we have $\mathbf{w} \cdot \mathbf{r} \leq k \cdot \mathsf{opt}$. At each iteration $\mathbf{w} \cdot \mathbf{r}^J$ becomes smaller by a factor of α, hence at the end of the step 3 loop we have

$$\mathbf{w} \cdot \mathbf{r}^J \leq (\mathbf{w} \cdot \mathbf{r}) \cdot \alpha^{\lceil \log_{1/\alpha} k \rceil} \leq \frac{\mathbf{w} \cdot \mathbf{r}}{k} \leq \frac{k \cdot \mathsf{opt}}{k} = \mathsf{opt} \ .$$

Consequently, we get that

$$\tau_{\mathbf{p}}(J \cup J') \leq \tau_{\mathbf{p}}(J) + \tau_{\mathbf{p}}(J') \leq \lceil \log_{1/\alpha} k \rceil (1+\gamma)\mathsf{opt} + 2\mathsf{opt} \ .$$

By choosing $\gamma = 2$ we get $\alpha = 1 - \frac{1}{2}(1 - \frac{1}{e})$, hence $\tau_{\mathbf{p}}(J \cup J') = O(\ln k)$, as required. This concludes the proof of Theorem 3.7.

We also mention that for unit/uniform powers, the problem admits a constant ratio.

Theorem 3.8 (Cohen & Nutov [12]) Min-Power Edge-Multi-Cover *with uniform powers admits a randomized approximation algorithm with expected approximation ratio ρ, where $\rho < 2.16851$ is the real root of the cubic equation $e(\rho - 1)^3 = 2\rho$.*

3.6 The Node-Weighted Steiner Forest problem

Let us recall the Node-Weighted Steiner Forest problem. We are given an undirected graph $G = (V, E)$ with node weights $\mathbf{w} = \{w_v : v \in V\}$ and a set R of demand pairs from V, and seek $J \subseteq E$ such that the graph (V, J) has an st-path for every demand pair $\{s, t\} \in R$. We want to minimize the node-weight of J, namely, the weight of the set of endnodes of the edges in J. We survey the proof of the following seminal result:

Theorem 3.9 (Klein & Ravi [21]) Node-Weighted Steiner Forest *admits ratio $2 \ln |U|$, where U is the union of the demand pairs.*

To prove Theorem 3.9 we use a ρ-**Density Algorithm** for the following generic problem:

Covering Problem
Input: Integral non-negative set functions ν, τ on a groundset E, where $\nu(\emptyset) > \nu(E) \geq 0$ and $\tau(\emptyset) = 0$.
Output: $J \subseteq E$ with $\nu(J) = \nu(E)$ and with $\tau(J)$ minimized.

A set function f is **increasing** if $f(A) \leq f(B)$ whenever $A \subseteq B$; f is **decreasing** if $-f$ is increasing, and f is **subadditive** if $f(A \cup B) \leq f(A) + f(B)$ for any two subset A, B of the groundset. It is easy to see that if f is subadditive (and non-negative) then f is increasing.

We call ν the **deficiency function** and τ a **payment function**; for a partial solution J, $\nu(J)$ measures how far is J from being feasible, while the function τ is our "payment" for J. In all our applications, the function ν is decreasing and τ is subadditive.

For a subset $S \subseteq E \setminus J$ the quantity $\sigma_J(S) = \dfrac{\tau(J \cup S) - \tau(J)}{\nu(J) - \nu(J \cup S)}$ is called the **density of** S w.r.t J. Let $\rho \geq \nu(E)+1$ be a parameter and let opt be the optimal solution value for an instance of a Covering Problem. The **ρ-Density Algorithm** starts with $J = \emptyset$, and as long as $\nu(J) > \nu(E)$, it adds to J an augmenting set $S \subseteq E \setminus J$ with $\nu(J \cup S) \leq \nu(J) - 1$ that satisfies the **ρ-Density Condition**. Since at each iteration $\nu(J \cup S) \leq \nu(J) - 1$, the algorithm terminates. It is easy to see that if ν is decreasing and τ is subadditive, then for any optimal solution J^*, the set $S = J^* \setminus J$ satisfies the ρ-Density Condition with $\rho = \nu(E) + 1$. Thus if $\nu(E)$ is small, then a low density set exists, and the problem is to find one in polynomial time. The following is implicitly proved in [19, 21].

Theorem 3.10 (Johnson [19], Klein & Ravi [21]) *The ρ-Density Algorithm computes a feasible solution J such that:* $\tau(J) \leq \rho \ln \dfrac{\nu(\emptyset)}{\nu(E)} \cdot \mathsf{opt}$ *if* $\nu(E) \geq 1$ *and* $\tau(J) \leq \rho(\ln \nu(\emptyset) + 1) \cdot \mathsf{opt}$ *if* $\nu(E) = 0$.

Proof: Let ℓ be the number of the iterations of the algorithm. Let J_i be the partial solution stored in J at the end of iteration i, and let $J_0 = \emptyset$. Let S_i be the set added to J_{i-1} at iteration i, so $J_i = J_{i-1} \cup S_i$, $i = 1, \ldots, \ell$. Since S_i satisfies the ρ-Density Condition w.r.t. J_{i-1}, we have $\dfrac{\tau(J_i) - \tau(J_{i-1})}{\nu(J_{i-1}) - \nu(J_i)} \leq \rho \cdot \dfrac{\mathsf{opt}}{\nu(J_{i-1})}$. Denote $\tau_i = \tau(J_i)$ and $\nu_i = \nu(J_i)$, where $\nu_0 = \nu(\emptyset)$. Then for every $i = 1, \ldots, \ell$

$$\nu_i \leq \nu_{i-1} \left(1 - \frac{\tau_i - \tau_{i-1}}{\rho \cdot \mathsf{opt}}\right)$$

Unraveling the last inequality gives that for any j with $\nu_j > 0$:

$$\frac{\nu_j}{\nu_0} \leq \prod_{i=1}^{j} \left(1 - \frac{\tau_i - \tau_{i-1}}{\rho \cdot \mathsf{opt}}\right).$$

Taking natural logarithm from both sides and using the fact that $\ln(1-x) \leq -x$ for $x < 1$ we obtain:

$$\ln\left(\frac{\nu_j}{\nu_0}\right) \leq \sum_{i=1}^{j} \ln\left(1 - \frac{\tau_i - \tau_{i-1}}{\rho \cdot \mathsf{opt}}\right) \leq -\sum_{i=1}^{j} \frac{\tau_i - \tau_{i-1}}{\rho \cdot \mathsf{opt}}.$$

Consequently, $\tau_j = \tau_j - \tau_0 = \sum_{i=1}^{j}(\tau_i - \tau_{i-1}) \leq \rho \ln \frac{\nu_0}{\nu_j} \cdot \mathsf{opt}$.

In the case $\nu(E) \geq 1$ we apply the last inequality for $j = \ell$ to get $\tau(J) = \tau_\ell \leq \rho \ln \frac{\nu_0}{\nu_\ell} \cdot \mathsf{opt}$, as required.

In the case $\nu(E) = 0$ we apply the last inequality for $j = \ell - 1$ to get $\tau_{\ell-1} \leq \rho \ln \frac{\nu_0}{\nu_{\ell-1}} \cdot \mathsf{opt} \leq \rho \ln \nu_0 \cdot \mathsf{opt}$. Observing that $\tau_\ell - \tau_{\ell-1} \leq \rho \cdot \mathsf{opt}$ we get $\tau(J) = \tau_\ell \leq \tau_{\ell-1} + \rho \cdot \mathsf{opt} \leq \rho(\ln \nu_0 + 1) \cdot \mathsf{opt}$, as required. □

If τ is subadditive, then $\tau(J \cup S) - \tau(J) \leq \tau(S)$, and thus we achieve the same performance as in Theorem 3.10 by replacing in the ρ-Density Algorithm the ρ-Density Condition by a stronger condition

$$\frac{\tau(S)}{\nu(J) - \nu(J \cup S)} \leq \rho \cdot \frac{\mathsf{opt}}{\nu(J)} \ .$$

In our setting of the Node-Weighted Steiner Forest problem, the groundset is the set E of edges, and for a partial solution J we define $\tau(J)$ to be the node-weight of J. It is easy to see that τ is subadditive.

The function ν is defined by $\nu(J) = \min\{|\mathcal{C}^J|, 1\}$, where $\mathcal{C}^J$ is the family of inclusionwise minimal **deficient sets** (a.k.a. tight sets) uncovered by J. More precisely, an undirected edge set J **covers** a node subset $A \subseteq V$ if J has an edge between A and $V \setminus A$, and $A \subseteq V$ is deficient if there exists a demand pair $\{s,t\}$ with $|\{s,t\} \cap A| = 1$. Then $\mathcal{C}^J$ is the family of deficient connected components of the graph (V, J). Note that $\nu(\emptyset) = |U|$, since when $J = \emptyset$ the minimal deficient sets are the singletons in U. It is easy to see that ν is decreasing and that J is a feasible solution iff $\nu(J) = 1$ (note that $|\mathcal{C}^J| \geq 1$ implies $|\mathcal{C}^J| \geq 2$).

With these definitions, we would like to apply the ρ-density Algorithm with $\rho = 2$ for our problem, namely, to design a polynomial time algorithm that for any partial solution J finds an augmenting edge set $S \subseteq E \setminus J$ of density $\leq 2\frac{\mathsf{opt}}{\nu(J)}$. We start by showing that there exists an augmenting edge set S of good density that has a simple structure.

Definition 3.4 *A* **spider** *is a rooted tree such that only its root, called the* **head***, may have degree ≥ 3. Equivalently, a spider is a union of paths that start at the same node – the head, such that no two paths have other node in common. A* **spider decomposition** *of a graph with a designated node subset U of terminals is a family $\mathcal{S}$ of node-disjoint spiders in the graph, with at least 2 terminals each, such that every terminal is a leaf or the head of some $S \in \mathcal{S}$.*

Lemma 3.6 (Klein & Ravi [21]) *Any tree T with a set U of at least 2 terminals has a spider decomposition.*

Proof: The proof is by induction on $|U|$. Root T at some node. Let v be a farthest node from the root such that the subtree of T that consists of v and its descendants contains at least 2 terminals. By the choice of v, the paths in the subtree from the terminals to v form a spider S with at least two terminals. Let T' be obtained from T by deleting the subtree rooted at v and let U' be the set of terminals in T'. If $U' = \emptyset$ we are done. If T' has a single terminal, then we joint to S the path from this terminal to v and obtain a spider that contains all terminals. Otherwise, by the induction hypothesis the pair T', U' admits a spider decomposition $\mathcal{S}'$. Then $\mathcal{S}' \cup \{S\}$ is a spider decomposition of T, U, as required. $\square$

Let J be a partial solution for a Node-Weighted Steiner Forest instance $G, R, \mathbf{w}$. We may consider the equivalent instance obtained from $G, R, \mathbf{w}$ by contracting into a single node of weight zero every nontrivial connected component of the graph (V, J), and updating accordingly the set R of demand pairs. Then for any $S \subseteq E \setminus J$, $\tau(J \cup S) - \tau(J)$ equals the weight of the endnodes of the edges in S in the new instance. Thus in what follows we will assume that $J = \emptyset$. Then the deficient sets are the singletons in U. We may also assume that the nodes in U have weight 0. For a subgraph S of G we denote by $S \cap U$ be the set of nodes in U that belong to S, but also use the notation $\nu(S)$ and $\tau(S)$ by considering S as the set of its edges. It is not hard to verify the following.

Lemma 3.7 *Let S be a connected subgraph of G with $|S \cap U| \geq 2$. Then $\nu(\emptyset) - \nu(S) = |S \cap U| - 1 \geq |S \cap U|/2$.*

Let us fix some inclusionwise minimal optimal solution F to the residual problem, so $\tau(F) \leq \mathsf{opt}$. Then F a forest, and any its non-trivial connected component is a tree with at least 2 terminals. Thus by Lemma 3.6 the pair F, U admits a spider decomposition $\mathcal{S}$. In the next two lemmas 3.8 and 3.9 we show that there is $S \in \mathcal{S}$ of low density, and how to find an edge set of density at most the density of any $S \in \mathcal{S}$.

Lemma 3.8 *There is a spider $S \in \mathcal{S}$ of density $\sigma_\emptyset(S) \leq 2\frac{\mathsf{opt}}{\nu(\emptyset)}$.*

Proof: Clearly, $\tau(F) \leq \mathsf{opt}$. Since the spiders in $\mathcal{S}$ are node disjoint $\sum_{S \in \mathcal{S}} \tau(S) \leq \mathsf{opt}$. Since the sets $S \cap U$ partition U and from Lemma 3.16 we have

$$\sum_{S \in \mathcal{S}} (\nu(\emptyset) - \nu(S)) \geq \sum_{S \in \mathcal{S}} |S \cap U|/2 = |U|/2 = \nu(\emptyset)/2 \ .$$

Consequently, by an averaging argument, there is $S \in \mathcal{S}$ of density $\sigma_\emptyset(S) = \frac{\tau(S)}{\nu(\emptyset) - \nu(S)} \leq \frac{\mathsf{opt}}{\nu(\emptyset)/2} = 2\frac{\mathsf{opt}}{\nu(\emptyset)}$, as claimed. $\square$

Now we show how to find in polynomial time an edge set S such that $\sigma_\emptyset(S) \leq \sigma_\emptyset(S')$ for any spider $S' \in \mathcal{S}$.

Lemma 3.9 *There exists a polynomial time algorithm that finds a subgraph S such that $\sigma_\emptyset(S) \leq \sigma_\emptyset(S^*)$, where S^* is a minimum density spider in $\mathcal{S}$.*

Proof: We may assume that we know the head h of S^* and the number $\ell = |S^* \cap U|$ (the "guess" of ℓ can be avoided, by using a slightly more complicated algorithm). Note that $\ell \geq 2$ and that $h \in U$ may hold. There is a polynomial number of choices, so we can try all choices and return the best outcome. The algorithm computes a set $P_1, \ldots, P_\ell$ of the lightest ℓ paths from h to a set of distinct nodes $u_1, \ldots, u_\ell$ in U (one of these nodes may be h, if $h \in U$), and returns their union S. It is easy to see that the algorithm can be implemented in polynomial time. We show that $\sigma_\emptyset(S) \leq \sigma_\emptyset(S^*)$. Let $P_1^*, \ldots, P_\ell^*$ be the ℓ paths from h to the terminals in S^*. Then

$$\tau(S) \leq \sum_{i=1}^{\ell} w(P_i) - (\ell-1)w_h \leq \sum_{i=1}^{\ell} w(P_i^*) - (\ell-1)w_h = \tau(S^*) \ .$$

By Lemma 3.7, $\nu(\emptyset) - \nu(S) \geq \ell - 1 = \nu(\emptyset) - \nu(S^*)$. Thus $\sigma_\emptyset(S) = \dfrac{\tau(S)}{\nu(\emptyset)-\nu(S)} \leq \dfrac{\tau(S^*)}{\nu(\emptyset)-\nu(S^*)} = \sigma_\emptyset(S^*)$, as claimed. □

Lemma 3.8 implies that the algorithm from Lemma 3.9 finds $S \subseteq E \setminus J$ of density at most $\sigma_J(S) \leq 2\dfrac{\mathsf{opt}}{\nu(J)}$. Thus we can find in polynomial time an edge set S obeying the ρ-Density Condition with $\rho = 2$. Since $\nu(\emptyset) = |U|$ and since $\tau = \tau_{\mathbf{w}}$ is subadditive, we get ratio $2 \ln |U|$ from Theorem 3.10.

3.7 The Min-Power k-Edge-Out-Connectivity problem

In this section we survey an $O(k \ln n)$-approximation algorithm for the directed Activation k-Edge-Out-Connectivity problem. For this, we will consider the augmentation variant of the problem, defined as follows.

Directed Activation k-Edge-Out-Connectivity Augmentation
Input: A directed graph $G_0 = (V, E_0)$ that is $(k-1)$-edge-out-connected from a given root node s, and an edge set E on V with power thresholds $\mathbf{p} = \{p_e : e \in E\}$.
Output: $J \subseteq E$ such that $G_0 \cup J$ is k-out-connected from s and $\tau_{\mathbf{p}}(J)$ is minimized.

It is easy to see that ratio ρ for the above augmentation version implies ratio $k\rho$ for directed Activation k-Edge-Out-Connectivity. In the rest of this section we describe the proof of the following result.

Theorem 3.11 (Nutov [28]) *The directed* Min-Power k-Edge-Out-Connectivity Augmentation *problem admits ratio* $3(\ln(n-1)+1)$.

Corollary 3.9 *Directed* Activation k-Edge-Connectivity *admits ratio* $O(k \ln n)$.

Proof: By Menger's Theorem, a directed graph is k-edge-connected iff for a given node s the graph it is both k-edge-out-connected from s and k-edge-in-connected to s. From this we get that ratio α for Activation k-Edge-Out-Connectivity and ratio β for Activation k-Edge-In-Connectivity implies ratio $\alpha+\beta$ for Activation k-Edge-Connectivity. From Theorem 3.11 we get $\alpha = 3k(\ln(n-1)+1)$ and from Corollary 3.2 we get $\beta = k$, and the ratio $O(k \ln n)$ follows. $\square$

Note that for $k = 1$ we have the directed Activation Out-Arborescence problem in Theorem 3.11 and the Activation Strong Connectivity problem in Corollary 3.9. For this particular case a slightly better ratio is known.

Theorem 3.12 (Calinescu, Kapoor, Olshevsky, & Zelikovsky [5]) Activation Out-Arborescence *admits ratio* $2(\ln(n-1)+1)$ *and* Activation Strong Connectivity *admits ratio* $2\ln(n-1)+3$.

3.7.1 Set-family edge-cover formulation

Let $\delta_J^{in}(A)$ denote the set of edges in J entering A. Let us say that a directed edge set J **covers** $A \subseteq V$ if J has an edge that enters A, namely, if $\delta_J^{in}(A) \neq \emptyset$. Given a set-family $\mathcal{F}$ we say that J **covers a set-family** $\mathcal{F}$ or that J is an **edge-cover of** $\mathcal{F}$ if every set in $\mathcal{F}$ is covered by some edge in J. The directed Activation k-Edge-Out-Connectivity Augmentation problem can be formulated as a particular case of the following problem.

Directed Activation Set-Family Edge-Cover
Input: A directed graph $G=(V,E)$ with power thresholds $\mathbf{p} = \{p_e : e \in E\}$ and a set-family $\mathcal{F}$ on V.
Output: An edge-cover $J \subseteq E$ of $\mathcal{F}$ such that $\tau_{\mathbf{p}}(J)$ is minimized.

We will assume that $\emptyset, V \notin \mathcal{F}$, as otherwise the problem has no feasible solution. In this problem, the family $\mathcal{F}$ may not be given explicitly, and for a polynomial time implementation of algorithms we just need that some queries related to $\mathcal{F}$ can be answered in polynomial time. The inclusionwise-minimal sets of a set-family $\mathcal{F}$ are called **$\mathcal{F}$-cores**, or just **cores**, if $\mathcal{F}$ is clear from the context. We denote the family of $\mathcal{F}$-cores by $\mathcal{C}(\mathcal{F})$. Given an edge set J let $\mathcal{F}^J$ denote the **residual family of** $\mathcal{F}$ (w.r.t. J), that consists of members of $\mathcal{F}$ not covered by J. We will assume that for any edge set J the family $\mathcal{C}(\mathcal{F}^J)$ of $\mathcal{F}^J$-cores can be computed in polynomial time. For our problem this can be done using a $n-1$ min-cut computations.

By Menger's Theorem, $J \subseteq E$ is a feasible solution to our problem iff J covers the family

$$\mathcal{F}_{\mathsf{OC}} = \{\emptyset \neq A \subseteq V \setminus \{s\} : |\delta_G^{in}(A)| = k-1\}$$

The following definition and lemma gives the essential property of the set-family $\mathcal{F}_{\mathsf{OC}}$ that we use.

Definition 3.5 *A set-family $\mathcal{F}$ on V with $\emptyset, V \notin \mathcal{F}$ is an* **intersecting family** *if $A \cap B, A \cup B \in \mathcal{F}$ holds for any $A, B \in \mathcal{F}$ that intersect.*

The following is known, c.f. [13].

Lemma 3.10 *$\mathcal{F}_{\mathsf{OC}}$ is an intersecting family.*

It is easy to see that the cores of an intersecting family are pairwise disjoint. Thus $|\mathcal{C}(\mathcal{F}_{\mathsf{OC}})| \leq n-1$. Hence to prove Theorem 3.11 it is sufficient to prove the following.

Theorem 3.13 (Nutov [28]) *Directed* Activation Set-Family Edge-Cover *with intersecting set-family $\mathcal{F}$ admits ratio* $3(\ln|\mathcal{C}(\mathcal{F})|+1)$.

For simplicity of exposition we give a proof of a slightly worse ratio $\frac{9}{2}(\ln|\mathcal{C}(\mathcal{F})|+1)$. We will again use a ρ-Density Algorithm for an appropriate Covering Problem. As before, E is the set of edges and $\tau(J)$ is the optimal value of an assignment activating J. The function ν is defined by $\nu(J) = |\mathcal{C}(\mathcal{F}^J)|$; note that J is a feasible solution iff $\nu(J) = 0$. To define an analogue of spiders, we study in the next section a special simple type of intersecting families.

3.7.2 Ring families

An intersecting set-family that has a unique core is called a **ring family**; equivalently, a set-family $\mathcal{F}$ with $\emptyset, V \notin \mathcal{F}$ is a **ring family** if $A \cap B, A \cup B \in \mathcal{F}$ for any $A, B \in \mathcal{F}$. Ring families often arise from intersecting families as follows.

Definition 3.6 *Let $\mathcal{F}$ be a set-family on V. For an $\mathcal{F}$-core $C \in \mathcal{C}(\mathcal{F})$ let $\mathcal{F}(C)$ denote the family of the sets in $\mathcal{F}$ that contain C and contain no $\mathcal{F}$-core distinct from C; for $h \in V$ let $\mathcal{F}(h, C) = \{A \in \mathcal{F}(C) : h \notin A\}$.*

Lemma 3.11 *Let $\mathcal{F}$ be an intersecting set-family on a groundset V.*

(i) *For any $\mathcal{F}$-core $C \in \mathcal{C}(\mathcal{F})$ and $h \in V$, $\mathcal{F}(h, C)$ is a ring family; in particular, $\mathcal{F}(C)$ is a ring family.*

(ii) *For any distinct $\mathcal{F}$-cores $C_i, C_j \in \mathcal{C}(\mathcal{F})$, no $A_i \in \mathcal{F}(C_i)$ and $A_j \in \mathcal{F}(C_j)$ intersect.*

Proof: We prove (i). Let $A, B \in \mathcal{F}(h, C)$. Then $A \cap B, A \cup B \in \mathcal{F}$, $C \subseteq A \cap B \subseteq A \cup B$, and $h \notin A \cup B \supseteq A \cap B$. It remains to prove is that $A \cup B$ contains no $\mathcal{F}$-core C' distinct from C. Otherwise, C' and one of A, B intersect, say $C' \cap A \neq \emptyset$. Then $C' \cap A \in \mathcal{F}$, hence by the minimality of C' we must have $C' \subseteq A$. This contradicts that $A \in \mathcal{F}(C)$.

We prove (ii). If $A_i \cap A_j \neq \emptyset$ then $A_i \cap A_j \in \mathcal{F}$, hence $A_i \cap A_j$ contains some $\mathcal{F}$-core C. This implies $C_i = C = C_j$, contradicting that C_i, C_j are distinct $\mathcal{F}$-cores. □

It is easy to see that if $\mathcal{F}$ is an intersecting or a ring family then so is the residual family $\mathcal{F}^J$ of $\mathcal{F}$, for any edge set J. In the following lemma we summarize the essential properties of ring families that we use.

Lemma 3.12 *Let J be an inclusionwise minimal directed edge-cover of a ring family $\mathcal{F}$ with core C. Then there is an ordering $e_1, \ldots, e_q$ of J and sets $C_1 \subset \cdots \subset C_q$ in $\mathcal{F}$ where $C_1 = C$, such that $\delta_J^{in}(C_i) = \{e_i\}$, and if $e_i = v_i u_i$ where $u_i \in C_i$, then $\{e_1, \ldots, e_i\}$ covers both $\mathcal{F}(v_i, C)$ and $\mathcal{F}(u_{i+1}, C)$. Furthermore, v_q does not belong to any set in $\mathcal{F}$.*

Proof: The proof of the main statement is by induction on $q = |J|$. For $q = 1$ the statement is obvious. If $|J| \geq 2$, let $e_1 \in \delta_J^{in}(C)$. Then $J' = J \setminus \{e_1\}$ is an inclusionwise minimal edge-cover of the residual family $\mathcal{F}' = \mathcal{F}^{\{e_1\}}$ of members of $\mathcal{F}$ not covered by e_1. By the induction hypothesis, there is an ordering $e_2, \ldots, e_q$

of J' and sets $C_2 \subset \cdots \subset C_q$ in $\mathcal{F}'$ as in the lemma. Since $C_1 = C$ is the unique $\mathcal{F}$-core, $C_1 \subset C_2$. To prove the lemma for $\mathcal{F}$ and J, we just need to show that e_2 does not cover C_1. Suppose to the contrary that $e_2 \in \delta_J^{in}(C_1)$. By the minimality of J, there is $A_1 \in \mathcal{F}$ such that $\delta_J(A_1) = \{e_1\}$. There is an edge in J covering $A_1 \cup C_2$, since $A_1 \cup C_2 \in \mathcal{F}$. This edge is one of e_1, e_2, since if an edge covers a union of two sets then it covers one of the sets. Each of e_1, e_2 covers $A_1 \cap C_2$, since $e_1, e_2 \in \delta_J^{in}(C_1)$ and $C_1 \subseteq A_1 \cap C_2$. Thus one of e_1, e_2 covers both $A_1 \cap C_2$ and $A_1 \cup C_2$. However, if an edge covers both $A \cap B, A \cup B$ then it covers both A and B. Hence one of e_1, e_2 covers both A_1, C_2. This contradicts our choice of A_1.

We show that there is no set $A \in \mathcal{F}$ with $v_q \in A$. Otherwise, all edges in J have both endnodes in $A \cup C_q$, hence $\delta_J^{in}(A \cup C_q) = \emptyset$. Since $\mathcal{F}$ is a ring family, $A \cup C_q \in \mathcal{F}$. This contradicts that J covers $\mathcal{F}$. □

Lemma 3.13 *The directed* Activation Set-Family Edge-Cover *problem with ring family $\mathcal{F}$ admits a polynomial time algorithm.*

Proof: It is known that computing a minimum cost edge cover of a ring family $\mathcal{F}$ can be done in polynomial time. Hence we can apply Corollary 3.1 with $\rho = 1$. From Lemma 3.12 it follows that if J is an inclusionwise minimal cover of $\mathcal{F}$ then $\Delta_J \leq 1$. Now the lemma follows from Corollary 3.1. □

3.7.3 Spider decompositions

To get some intuition, let us first restate some concepts from Section 3.6 in terms of directed graphs. A **(directed) spider** is a union of directed paths that start at the same node such that no two paths have other node in common. We say that a spider **hits** a node u if u is a leaf or the head of S; a family $\mathcal{S}$ of spiders hits a node set U if every $u \in U$ is hit by some $S \in \mathcal{S}$. Recall that Lemma 3.6 states that any (undirected) tree T with a set of terminals has a spider decomposition – a family $\mathcal{S}$ of node-disjoint spiders, such that every $S \in \mathcal{S}$ hits at least 2 terminals, and such that $\mathcal{S}$ hits all terminals. Here we consider spider decompositions of directed graphs when the spiders should still be node-disjoint, but the other two conditions are relaxed. Spiders that hit just one terminal are allowed, but should satisfy a certain condition. Also, spiders may not hit all terminals, but just some fraction of them. For example, by the same proof as in Lemma 3.6, one can prove the following lemma, that allows spiders hitting just one terminal.

Lemma 3.14 *Any out-arborescence T with a set U of terminals and root s contains a family $\mathcal{S}$ of node-disjoint spiders that hits U, such that any $S \in \mathcal{S}$ that contains a single terminal u is the su-path in T.*

In this section we define spiders related to set families, and then state and prove an appropriate spider-decomposition theorem of edge-covers of intersecting families. Let us say that two edge sets S and S' are V'-disjoint if no $e \in S$ and $e' \in S'$ have a common endnode in V'.

Definition 3.7 *Let $\mathcal{F}$ be a set-family on V. An $\mathcal{F}$-**spider** is a triple $h, \mathcal{C}, S$, where $h \in V$ is the **head** of the $\mathcal{F}$-spider, $\mathcal{C} \subseteq \mathcal{C}(\mathcal{F})$ is the set of cores **hit by** the $\mathcal{F}$-spider, and S is an edge set that is a union of (possibly empty) pairwise $(V \setminus \{h\})$-disjoint $\mathcal{F}(h, C)$-covers $\{S_C : C \in \mathcal{C}\}$, such that if $\mathcal{C} = \{C\}$ then h does not belong to any set in $\mathcal{F}(C)$. We will often denote an $\mathcal{F}$-spider just by S, meaning that there exists an appropriate choice of h and $\mathcal{C}$.*

The purpose of this section is to prove the following.

Theorem 3.14 (Nutov [28]) *Any directed cover J of an intersecting family $\mathcal{F}$ with ℓ cores contains a family $\mathcal{S}$ of node-disjoint $\mathcal{F}$-spiders that hits at least $\frac{2}{3}\ell$ distinct cores in $\mathcal{C}(\mathcal{F})$.*

Note that the spider decomposition in the theorem differs from the one in Lemma 3.6 in two ways: an $\mathcal{F}$-spider $S \in \mathcal{S}$ may hit just one $\mathcal{F}$-core, and the spiders in $\mathcal{S}$ may not hit all $\mathcal{F}$-cores.

A simple proof of Theorem 3.14 relies on a spider decomposition of families of directed paths.

Definition 3.8 *Let $\mathcal{P}$ be a family of simple directed paths with a set $U(\mathcal{P})$ of distinct endnodes. We say that a spider S is a $\mathcal{P}$-**spider** if S is a union of internally-disjoint (h, U)-subpaths (possibly of length 0) of the paths in $\mathcal{P}$, for some $U \subseteq U(\mathcal{P})$ and a node h called the **head of** S (possibly $h \in U$), such that if $|U| = 1$ then $S \in \mathcal{P}$.*

We will need the following spider decomposition lemma that was implicitly proved in [28].

Lemma 3.15 ([28]) *Let $\mathcal{P}$ be a family of ℓ simple directed paths in a graph G with a set $U(\mathcal{P})$ of distinct endnodes. If G is an out-arborescence then G contains a family of node-disjoint $\mathcal{P}$-spiders that hits $U(\mathcal{P})$. If G has maximum indegree 1 then G contains a family of node-disjoint $\mathcal{P}$-spiders that hits at least $\frac{2}{3}\ell$ nodes in $U(\mathcal{P})$.*

Proof: We prove the arborescence case by induction on ℓ. If $\ell = 1$ then $\mathcal{S} = \mathcal{P}$ is a family as required. Suppose that $\ell \geq 2$. If there is a path P that has no node in common with other paths then the induction is obvious. Otherwise, let h be a farthest node from the root such that the subtree of G induced by h and its descendants hits at least two nodes in $U(\mathcal{P})$. By the choice of h, in the subtree, the paths from h to the nodes in $U(\mathcal{P})$ form a $\mathcal{P}$-spider that hits at least 2 nodes in $U(\mathcal{P})$. Let G' be obtained from G by removing this subtree and let $\mathcal{P}'$ be obtained by removing from $\mathcal{P}$ all the paths that have a node in this subtree. If $\mathcal{P}' = \emptyset$ then $\mathcal{S} = \{S\}$ is a family of $\mathcal{P}$-spiders as required. Otherwise, by the induction hypothesis there exists a family $\mathcal{S}'$ of node-disjoint spiders for $\mathcal{P}'$ as in the lemma. Then $\mathcal{S} = \mathcal{S}' \cup \{S\}$ is a family of $\mathcal{P}$-spiders as required. This concludes the proof of the arborescence case.

Suppose that G has maximum indegree 1. Then G is a collection of node-disjoint directed graphs of the following type: each of the graphs is a cycle (that may be a single node) with node-disjoint arborescences (that may be single nodes) attached to the cycle by the roots. Since these graphs are node-disjoint, it is sufficient to consider the case when G is such a graph. If G is acyclic, then G is an out-arborescence and we are done. If $\ell \geq 3$ then we arrive at the arborescence case by removing one edge from the cycle of G and removing the path that contains this edge from $\mathcal{P}$ – then we get a family $\mathcal{S}$ of node-disjoint spiders that hits least $\ell - 1 \geq \frac{2}{3}\ell$ nodes in $U(\mathcal{P})$. The remaining case is $\ell = 2$, say $\mathcal{P} = \{P_1, P_2\}$ and $U(\mathcal{P}) = \{u_1, u_2\}$. Let h be a common node of P_1 and P_2. Then the union of the hu_1-subpath of P_1 and the hu_2-subpath of P_2 is a $\mathcal{P}$-spider that hits all nodes in $U(\mathcal{P})$. $\square$

We note that Lemma 3.15 was extended by Chuzhoy and Khanna [11] to an arbitrary graph G, but the proof of the case when G has maximum indegree 1 is simpler, and Lemma 3.15 suffices for the proof of Theorem 3.14.

Now we use Lemmas 3.15 and 3.12 to prove Theorem 3.14.

Proof of Theorem 3.14: For every $C \in \mathcal{C}(\mathcal{F})$ fix some inclusionwise-minimal edge-cover $J_C \subseteq J$ of $\mathcal{F}(C)$. By lemma 3.11(i), $\mathcal{F}(C)$ is a ring family. Let $e_1, \ldots, e_q$ be an ordering of J_C and $C_1 \subset \cdots \subset C_q$ sets in $\mathcal{F}(C)$ as in Lemma 3.12, where $e_i = v_i u_i$ is as in the lemma. Obtain a directed path P_C by adding for every $i = q, \ldots, 2$ the directed edge $u_i v_{i-1}$, if $u_i \neq v_{i-1}$; e.g., if $u_i \neq v_{i-1}$ for all i, then the node sequence of

P_C is $(v_q, u_q, v_{q-1}, u_{q-1}, \ldots, v_1, u_1)$. Denote $M_C = C_q$ and $u_C = u_1$, and note that $u_C \in C \subseteq M_C \in \mathcal{F}(C)$.

Let $\mathcal{P} = \{P_C : C \in \mathcal{C}(\mathcal{F})\}$. Each directed path $P_C \in \mathcal{P}$ has only its starting node outside M_C, and the sets $\{M_C : C \in \mathcal{C}(\mathcal{F})\}$ are pairwise disjoint, by Lemma 3.11(ii). Thus any two paths in $\mathcal{P}$ have distinct endnodes and the union of the paths in $\mathcal{P}$ is a graph of maximum indegree 1. Hence Lemma 3.15 applies, and there exists a family $\hat{S}$ of node-disjoint $\mathcal{P}$-spiders that hits at least $\frac{2}{3}\ell$ nodes in $U(\mathcal{P}) = \{u_C : C \in \mathcal{C}(\mathcal{F})\}$.

Consider a $\mathcal{P}$-spider $\hat{S} \in \hat{\mathcal{S}}$. Let h be the head of $\hat{S}$ and let $\{u_C : C \in \mathcal{C}\}$ be the set of nodes in $U(\mathcal{P})$ hit by this spider. Let $S = \hat{S} \cap J$ be the edge set obtained from $\hat{S}$ by removing the added edges. To prove the theorem it is sufficient to show that the triple $h, \mathcal{C}, S$ is an $\mathcal{F}$-spider. For $C \in \mathcal{C}$ let $\hat{S}_C$ be the hu_C-path in $\hat{S}$ and let $S_C = \hat{P}_C \cap J$. Since $\hat{S}$ is a spider, the edge sets $\{S_C : C \in \mathcal{C}\}$ are pairwise $(V \setminus \{h\})$-disjoint. By Lemma 3.12, each S_C is an $\mathcal{F}(h, C)$-cover. Furthermore, if $\mathcal{C} = \{C\}$ then $\hat{S}_C = P_C$, since $\hat{S}$ is a $\mathcal{P}$-spider. This implies that $S_C = J_C$ and $h = v_q$. By Lemma 3.12, v_q does not belong to any set in $\mathcal{F}(C)$. Thus the triple $h, \mathcal{C}, S$ is an $\mathcal{F}$-spider, and the proof is complete. $\square$

3.7.4 Finding a low density $\mathcal{F}$-spider

Let J be a partial solution for a directed Activation Set-Family Edge-Cover instance $G, \mathbf{p}, \mathcal{F}$. We may consider the equivalent instance obtained by removing J from G and replacing $\mathcal{F}$ by the residual family $\mathcal{F}^J$. Clearly, the optimal solution value of the new instance is at most the optimal solution value of the original instance. Thus in what follows we will assume that $J = \emptyset$. In the following lemma we lower bound the decrease of the deficiency function caused by a union of $\mathcal{F}(h, C)$-covers $\{S_C : C \in \mathcal{C}\}$, and in particular by an $\mathcal{F}$-spider.

Lemma 3.16 *Let $\mathcal{F}$ be an intersecting set-family on V, let $\mathcal{C} \subseteq \mathcal{C}(\mathcal{F})$, let $h \in V$, and let S be a directed edge set that covers $\mathcal{F}(h, C)$ for every $C \in \mathcal{C}$, such that if $\mathcal{C} = \{C\}$ then h does not belong to any set in $\mathcal{F}(C)$. Then $\nu(\emptyset) - \nu(S) \geq |\mathcal{C}|/3$.*

Proof: The $\mathcal{F}^S$-cores are pairwise disjoint and each of them contains some $\mathcal{F}$-core. Let t be the number of $\mathcal{F}^S$-cores that contain exactly one $\mathcal{F}$-core. Any other $\mathcal{F}^S$-core contains at least two $\mathcal{F}$-cores. Thus $\nu(\emptyset) - \nu(S) \geq \lceil (\nu(\emptyset) - t)/2 \rceil$. We upper bound t as follows. By the definition of S, any $\mathcal{F}^S$-core C' that contains some $C \in \mathcal{C}$, contains h or contains some $\mathcal{F}$-core distinct from C. Furthermore, if $\mathcal{C} = \{C\}$, then the latter must hold. As the $\mathcal{F}^S$-cores are pairwise disjoint, h belongs to at most one of them. Thus $t \le \nu(\emptyset) - (|\mathcal{C}| - 1)$ if $|C| \ge 2$, and $t \le \nu(\emptyset) - 1$ if $|\mathcal{C}| = 1$. In both cases we have $\nu(\emptyset) - \nu(S) \ge |\mathcal{C}|/3$. □

In [28] a better bound $\nu(\emptyset) - \nu(S) \ge |\mathcal{C}|/2$ is established under additional assumptions on S. This is why the ratio $\frac{9}{2}(\ln|\mathcal{C}(\mathcal{F})| + 1)$ proved here is worse by a factor $\frac{3}{2}$ than the ratio $3(\ln|\mathcal{C}(\mathcal{F})| + 1)$ in [28].

The following lemma shows that there exists an $\mathcal{F}$-spider of low density.

Lemma 3.17 *Let $\mathcal{S}$ be a family of $\mathcal{F}$-spiders as in Theorem 3.14 for an optimal directed cover of an intersecting family $\mathcal{F}$. There is an $\mathcal{F}$-spider $S^*, \mathcal{C}^*, h$ in $\mathcal{S}$ of such that* $\dfrac{\tau(S^*)}{|\mathcal{C}^*|/3} \le \dfrac{9}{2} \cdot \dfrac{\mathsf{opt}}{\nu(\emptyset)}$.

Proof: Let $\mathcal{C}_S$ denote the set of $\mathcal{F}$-cores hit by a spider $S \in \mathcal{S}$. Since the spiders in $\mathcal{S}$ are node disjoint $\sum_{S \in \mathcal{S}} \tau(S) \le \mathsf{opt}$. Since $\mathcal{S}$ hits at least $\dfrac{2}{3}\nu(\emptyset)$ distinct $\mathcal{F}$-cores $\sum_{S \in \mathcal{S}} |\mathcal{C}_S| \ge \dfrac{2}{3}\nu(\emptyset)$. Thus $\sum_{S \in \mathcal{S}} |\mathcal{C}_S|/3 \ge \dfrac{2}{9}\nu(\emptyset)$. Consequently, by an averaging argument, there is $S^* \in \mathcal{S}$ as required. □

Lemma 3.18 *There exists a polynomial time algorithm that given an instance of directed* Activation Set-Family Edge-Cover *with intersecting set-family $\mathcal{F}$ finds an edge set $S \subseteq E$ of density* $\dfrac{\tau(S)}{\nu(\emptyset) - \nu(S)} \le \dfrac{9}{2} \cdot \dfrac{\mathsf{opt}}{\nu(\emptyset)}$.

Proof: Let $S^*, \mathcal{C}^*, h$ be a spider as in Lemma 3.17. As in the proof of Lemma 3.9, we may assume that we know h, the power level w_h of h in S^*, and the number $\ell = |\mathcal{C}^*|$. Then the algorithm is as follows.

Algorithm 7: LOW-DENSITY $\mathcal{F}$-SPIDER$(G = (V, E), \mathbf{p}, \mathcal{F}, h, w_h, \ell)$

1 for every edge $hv \in E$ do: $p_{hv} \leftarrow 0$ if $p_{hv} \le w_h$ and $E \leftarrow E \setminus \{hv\}$ otherwise

2 for every core $C \in \mathcal{C}(\mathcal{F})$ compute an optimal $\mathcal{F}(h, C)$-cover P_C

3 if $\ell = 1$ then return $S = \arg\min\{\tau(P_C) : C \in \mathcal{C}(\mathcal{F}), \nu(P_C) \le \nu(\emptyset) - 1\}$

4 else return the union S of ℓ lowest value sets P_C

The algorithm can be implemented in polynomial time using the algorithm from Lemma 3.13. We show that $\tau(S) \le \tau(S^*)$. Let $\tau'(J)$ denote the optimal assignment value activating J with the modified power thresholds after step 1. Then $\tau'(S) \le \tau'(S^*)$, since S^* is a union of ℓ pairwise $(V \setminus \{h\})$-disjoint $\mathcal{F}(h, C)$ covers while S is a union of ℓ lowest τ'-value $\mathcal{F}(h, C)$-covers. Also, $\tau(S) \le \tau'(S) + w_h$ while $\tau(S^*) = \tau'(S^*) + w_h$. Thus we get $\tau(S) \le \tau'(S) + w_h \le \tau'(S^*) + w_h = \tau(S^*)$. Consequently, from Lemma 3.16 and our choice of S^* we will get $\dfrac{\tau(S)}{\nu(\emptyset) - \nu(S)} \le \dfrac{\tau(S^*)}{\ell/3} \le \dfrac{9}{2} \cdot \dfrac{\mathsf{opt}}{\nu(\emptyset)}$, as required. □

Lemma 3.17 implies that the algorithm from Lemma 3.18 finds and edge set $S \subseteq E \setminus J$ of density $\leq \frac{9}{2} \cdot \frac{\mathsf{opt}}{\nu(\emptyset)}$; namely, J satisfies the ρ-Density Condition with $\rho = \frac{9}{2}$. Thus we can apply the ρ-Density Algorithm with $\rho = \frac{9}{2}$. Since $\nu(\emptyset) = |\mathcal{C}(\mathcal{F})|$ we get ratio $\frac{9}{2}(\ln|\mathcal{C}(\mathcal{F})| + 1)$ from Theorem 3.10.

3.8 Open problems

In this section we list some open problems in the field, most of them for the case of high demands.

The undirected Min-Power Edge-Multi-Cover problem. The currently best known ratio for the problem is $\min\{O(\ln k), k + 1/2\}$, while for unit/uniform powers a constant ratio is known [12]. A constant ratio for the problem would imply several consequences, via Corollary 3.5. For example, we would get a constant ratio for the Min-Power k-Out-Connectivity problem. More importantly, we get that for the k-Connectivity problem, the approximability of the min-cost and the min-power versions differs by a constant factor. It is an old open problem whether the Min-Cost k-Connectivity problem admits a constant ratio, and relation between the min-cost and the min-power variants might help to resolve it.

Problems with linear ratios. For several min-power and node-weighted problems the currently best known ratio is $O(k)$, or even $O(k \ln n)$. The simplest examples are directed/undirected Min-Power k Edge-Disjoint Paths, and directed Min-Power k-Connectivity and Min-Power k-Edge-Connectivity. As was mentioned in the Introduction, these problems are unlikely to admit polylogarithmic ratios [24, 18]. However, this does not exclude ratios sublinear in k. The simplest open problem is whether directed or undirected k Edge-Disjoint Paths problems admits ratio $k^{1-\epsilon}$ for some $\epsilon > 0$.

Problems with undetermined complexity status. The best known ratio for the min-power and node-weighted undirected k Disjoint Paths problem is 2. However, the problem is not known to be NP-hard. Does the problem admits a polynomial time algorithm? A related (and probably easier) question is whether the problem of covering a ring set-family by undirected edges admits a polynomial time algorithm for min-power or node-weighted setting. The currently best known ratio for this problem is also 2.

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] 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.

[3] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *STOC*, pages 201–210, 2010.

[4] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanitá. Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1):6, 2013.

[5] G. Calinescu, S. Kapoor, A. Olshevsky, and A. Zelikovsky. Network lifetime and power assignment in ad hoc wireless networks. In *ESA*, pages 114–126, 2003.

[6] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos. Energy-efficient wireless network design. *Theory Comput. Syst.*, 39(5):593–617, 2006. Preliminary version in ISAAC 2003, 585-594.

[7] W. Chen and N. Huang. The strongly connecting problem on multihop packet radio networks. *IEEE Transactions on Communications*, 37(3):293295, 1989.

[8] J. Cheriyan, T. Jordán, and Z. Nutov. On rooted node-connectivity problems. *Algorithmica*, 30(3):353–375, 2001.

[9] J. Cheriyan and R. Thurimella. Approximating minimum-size k-connected spanning subgraphs via matching. *SIAM J. Computing*, 30(2):528–560, 2000.

[10] J. Cheriyan and L. Végh. Approximating minimum-cost k-node connected subgraphs via independence-free graphs. *SIAM J. Computing*, 43(4):1342–1362, 2014.

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

[12] N. Cohen and Z. Nutov. Approximating minimum power edge-multi-covers. *J. Comb. Optim.*, 30(3):563–578, 2015.

[13] A. Frank. *Connections in Combinatorial Optimization.* Oxford University Press, 2011.

[14] A. Frank and E. Tardos. An application of submodular flows. *Linear Algebra and its Applications*, 114/115:329–348, 1989.

[15] T. Fukunaga, Z. Nutov, and R. Ravi. Iterative rounding approximation algorithms for degree-bounded node-connectivity network design. *SIAM J. Computing*, 44(5):1202–1229, 2015.

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

[17] F. Grandoni. On min-power steiner tree. In *ESA*, pages 527–538. 2012.

[18] M. Hajiaghayi, G. Kortsarz, V. Mirrokni, and Z. Nutov. Power optimization for connectivity problems. *Math. Program.*, 110(1):195–208, 2007.

[19] D. Johnson:. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.*, 9(3):256–278, 1974.

[20] S. Khuller and U. Vishkin. Biconnectivity approximations and graph carvings. *Journal of the Association for Computing Machinery*, 41(2):214–235, 1994.

[21] P. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *J. Algorithms*, 19(1):104–115, 1995.

[22] G. Kortsarz, V. Mirrokni, Z. Nutov, and E. Tsanko. Approximating minimum-power degree and connectivity problems. *Algorithmica*, 60(4):735–742, 2011.

[23] G. Kortsarz and Z. Nutov. Approximating minimum-power edge-covers and $2, 3$-connectivity. *Discrete Applied Mathematics*, 157(8):1840–1847, 2009.

[24] Y. Lando and Z. Nutov. On minimum power connectivity problems. *J. Discrete Algorithms*, 8(2):164–173, 2010.

[25] W. Mader. Ecken vom grad n in minimalen n-fach zusammenhängenden graphen. *Archive der Mathematik*, 23:219–224, 1972.

[26] Z. Nutov. The k-connected subgraph problem. In T. Gonzalez, editor, *Approximation Algorithms and Metaheuristics*, chapter ?? Chapman & Hall, 20??

[27] Z. Nutov. Node-connectivity survivable network problems. In T. Gonzalez, editor, *Approximation Algorithms and Metaheuristics*, chapter ?? Chapman & Hall, 20??

[28] Z. Nutov. Approximating minimum power covers of intersecting families and directed edge-connectivity problems. *Theor. Comput. Sci.*, 411(26-28):2502–2512, 2010.

[29] Z. Nutov. Approximating minimum-power k-connectivity. *Ad Hoc & Sensor Wireless Networks*, 9(1-2):129–137, 2010.

[30] Z. Nutov. Approximating steiner networks with node-weights. *SIAM J. Comput.*, 39(7):3001–3022, 2010.

[31] Z. Nutov. Survivable network activation problems. *Theor. Comput. Sci.*, 514:105–115, 2013.

[32] Z. Nutov. Approximating minimum-cost edge-covers of crossing biset-families. *Combinatorica*, 34(1):95–114, 2014.

[33] D. Panigrahi. Survivable network design problems in wireless networks. In *SODA*, pages 1014–1027. 2011.

[34] A. Schrijver. *Combinatorial Optimization, Polyhedra and Efficiency.* Springer-Verlag,

[35] E. Tsanko. M.sc. thesis: Approximating minimum power network design problems. Dept. of Computer Science, The Open University of Israel, 2006.

Index