Approximating Minimum-Power Network Design Problems

Guy Kortsarz * Vahab Mirrokni† Zeev Nutov‡ Elena Tsanko§

Abstract

Given a graph with costs on the edges, the power of a node is the maximum cost of an edge leaving it, and the power of the graph is the sum of the powers of its nodes. Motivated by applications in wireless networks, we consider several network design problems under the power minimization criteria. Given a graph $\mathcal{G} = (V, \mathcal{E})$ with costs on the edges and requirements $r(v)$ for each $v \in V$, the *Min-Power Edge-Multi-Cover* problem (MPEMC) is to find a min-power subgraph so that the degree of every node v is at least $r(v)$. We give an $O(\log n)$-approximation algorithms for MPEMC (improving the previously best known $O(\log^4 n)$-approximation [17]); this implies an $O(\log n+\alpha)$-approximation algorithm for the undirected *Min-Power k-Connected Subgraph* (MPk-CS) problem, where α is the best known approximation for the min-cost variant of the problem. (Currently, $\alpha = O(\ln k)$ for $n \geq 2k^2$ and $\alpha = O(\ln^2 k \cdot \min\{\frac{n}{n-k}, \frac{\sqrt{k}}{\ln n}\})$ otherwise.) We also consider the case of small requirements. Specifically, some of our approximation ratios are: 3/2 for MPEMC with $r(v) \in \{0, 1\}$ (improving the ratio 2 by [17]) and $3\frac{2}{3}$ (improving the ratio 4 by [6]) for the min-power 2-connected and 2-edge-connected spanning subgraph problems. Finally, we give a $4r_{\max}$-approximation algorithm for the undirected min-power Steiner Network problem: find a min-power subgraph that contains $r(u, v)$ pairwise edge-disjoint paths for every pair u, v of nodes.

*Rutgers University, Camden, guyk@camden.rutgers.edu

†Microsoft Research, mirrokni@csail.mit.edu

‡The Open University of Israel, Raanana nutov@openu.ac.il

§The Open University of Israel, elena@openu.ac.il

1 Introduction

1.1 Motivation

Wireless networks are an important subject of study due to their extensive applications. A large research effort focused on designing fault-tolerant networks while minimizing the power consumption of the stations of the network. The power consumption of a station determines its transmission range, and thus also the stations it can send messages to; the power typically increases at least quadradically in the transmission range. Assigning power levels to the stations (nodes) determines the resulting communication network. Conversely, a given communication network incures at every node v only a cost of a direct transmission to the farhest node from v. In contrast, in wired networks every two stations that need to communicate directly incure a cost. We study the design of symmetric wireless networks that meet some prescribed connectivity or degree properties, and such that the total power is minimized.

An important network property is fault-tolerance, which is often measured by node-connectivity or the degrees of the nodes in the network. These variants of fault-tolerant power-minimization problems were already extensively studied, c.f., [2, 5, 16, 17, 9]. The simplest connectivity problem is where we require the network to be connected. In this case, the min-cost variant is just the min-cost spanning tree problem, while the min-power variant is APX-hard [9]. A 5/3-approximation algorithm for the min-power spanning tree problem is given in [2].

1.2 Notation and basic definitions

Unless stated otherwise the graphs are assumed to be undirected. Let $\mathcal{G} = (V, \mathcal{E}; c)$ be a *network*, that is $(V, \mathcal{E})$ is a (possibly directed) graph and c is a cost function on $\mathcal{E}$. Let $n = |V|$ and $m = |\mathcal{E}|$. We sometimes write $\mathcal{G} = (V, \mathcal{E})$ and refer to $\mathcal{G}$ as graph. Let $G = (V, E)$ be a spanning subgraph of $\mathcal{G}$. For $v \in V$, the *power* $p(v) = p_c(v)$ *of* v in G (w.r.t. c) is the maximum cost of an edge in G incident to v. In directed graphs, $p(v)$ is the maximum cost of an edge leaving v, while the edges entering v do not affect its power. The power of the graph is the sum of the powers of its nodes.

For disjoint $X, Y \subseteq V$ let $\delta_G(X, Y) = \delta_E(X, Y)$ be the set of edges from X to Y in E, and let $d(X, Y) = |\delta_G(X, Y)|$ be the number of edges in G going from X to Y. We sometimes omit the subscripts G and E if they are clear from the context. For brevity, $\delta_E(X) = \delta_E(X, V - X)$, and $d_E(X) = |\delta_E(X)|$ is the *degree of* X.

1.3 Problem Formulation

Given a network $\mathcal{G} = (V, \mathcal{E}; c)$, our goal is to find a low power *communication network*, that is, a low power subgraph $G = (V, E)$ of $\mathcal{G}$ that satisfies some prescribed property. A fundamental property is to satisfy prescribed degree requirements. Given an integral requirement function r on V, we say that G (or E) is an r-*edge cover* if $d_G(v) \geq r(v)$ for every $v \in V$. In the case of directed graphs we require that the indegree of v in G is at least $r(v)$. We consider the following fundamental problem:

Min-Power Edge-Multi-Cover (MPEMC):
Instance: A network $\mathcal{G} = (V, \mathcal{E}; c)$ and requirements $\{r(v) : v \in V\}$.

Objective: Find a min-power subgraph $G = (V, E)$ of $\mathcal{G}$ so that G is an r-edge cover.

The budgeted version of the problem is:

Power Budgeted Maximum Edge-Multi-Coverage (PBMEMC):
Instance: A network $\mathcal{G} = (V, \mathcal{E}; c)$ with requirements $\{r(v) : v \in V\}$ and a power budget P.
Objective: Find a subgraph G of $\mathcal{G}$ with $p(G) \leq P$ and maximum $\sum_{v \in V} \min\{d_G(v), r(v)\}$.

MPEMC naturally arises in applications. For example, given designated sets A of "suppliers" and B of "clients" (A, B may not be disjoint), we seek to design a low power communication network in which every client can communicate with at least one supplier. The fault-tolerance variant of this scenario requests reliability of communication: each $b \in B$ needs to be able to communicate with at least $r(b)$ suppliers from A.

A (simple) graph is k-node-connected if it contains k internally-disjoint uv-paths between every pair u, v of its nodes. In power optimization, it is more natural to study crashes of nodes rather than edges. A central min-power fault-tolerance connectivity problem is:

Min-Power k-Connected Subgraph (MPk-CS):
Instance: A graph $\mathcal{G} = (V, \mathcal{E})$ with costs on the edges, and an integer k.
Objective: Find a min-power vertex k-connected spanning subgraph G of $\mathcal{G}$.

In the edge-connectivity variant **Min-Power k-Edge Connected Subgraph** (MPk-ECS) problem, the paths are required to be only edge disjoint. We study a generalization:

Min-Power Steiner Network (MPSN):
Instance: A network $\mathcal{G} = (V, \mathcal{E}; c)$ and requirement $r(u, v)$ for every node pair $u, v \in V$.
Objective: Find a min-power subgraph G of $\mathcal{G}$ so that G contains $r(u, v)$ pairwise edge-disjoint uv-paths for every $u, v \in V$.

Our main results are for the undirected case, but sometimes we will need to discuss min-power problems on directed networks. Note that in this case $p(v)$ is the maximum cost of an edge *leaving* v, while the degree requirements are on the number of edges *entering* v. For example, in MPEMC the indegree of a node v should be at least $r(v)$, while in PBMEMC we want to maximize $\sum_{v \in V} \min\{d_G^+(v), r(v)\}$, where $d_G^+(v)$ is the indegree of v in G.

Given an instance of a problem, we assume that a feasible solution exists; otherwise our algorithms can be easily modified to return an error message. Let opt denote the optimal solution value of an instance at hand.

1.4 Previous work

Previous results on MPEMC: The Min-Cost Edge Multicover problem is solvable in polynomial time c.f., [10], while the min-power variant MPEMC is APX-hard [17]. The previously best known approximation ratio for MPEMC is $\min\{r_{\max} + 1, O(\log^4 n)\}$ due to Hajiaghayi et. al [17]. It is interesting to note that the directed MPEMC generalizes the classic Min-Cost Set-Multicover problem; the later is a particular case where for every node $v \in V$ the costs of the edges leaving v are the same. In the same way directed PBMEMC generalizes the Cost-Budgeted Maximum Coverage problem, that admits a $(1 - 1/e)$-approximation algorithm [20], which is tight unless P=NP. The algorithm of [1] can be used to approximate the directed PBMEMC problem within ratio $(1-1/e)$ and the directed MPEMC within ratio $H(\Delta)$, where Δ is the maximum outdegree of a node in $\mathcal{G}$, and $H(k)$ denotes the kth Harmonic number. The details are omitted in this extended abstract.

Previous results on connectivity problems: Min-cost connectivity problems were extensively studied, see surveys in [19] and [24]. The best known approximation ratios for the **Min-Cost k-Connected Subgraph** (MCk-CS) problem are $O(\ln^2 k \cdot \min\{\frac{n}{n-k}, \frac{\sqrt{k}}{\ln k}\})$ for both directed and undirected graphs [23], and $O(\ln k)$ for undirected graphs with $n \geq 2k^2$ [8]. It turns out that (for undirected graphs) approximating MPk-CS is closely related to approximating MCk-CS and MPEMC, as shows the following statement.

Theorem 1.1 ([17])
(i) *If there exists an α-approximation algorithm for* MCk-CS *and a β-approximation algorithm for* MPEMC *then there exists a $(2\alpha+\beta)$-approximation algorithm for* MPk-CS.
(ii) *If there exists a ρ-approximation for* MPk-CS *then there exists a $(2\rho+1)$-approximation for* MCk-CS.

One can combine various values of α, β with Theorem 1.1 to get approximation algorithms for MPk-CS. In [17] the bound $\beta = \min\{k+1, O(\log^4 n)\}$ was derived. The best known values for α are: $\alpha = \lceil (k+1)/2 \rceil$ for $2 \leq k \leq 7$ (see [3] for $k = 2, 3$, [11] for $k = 4, 5$, and [22] for $k = 6, 7$); $\alpha = k$ for $k = O(\log n)$ [22], $\alpha = 6H(k)$ for $n \geq k(2k-1)$ [8], and $\alpha = O(\ln k \cdot \min\{\sqrt{k}, \frac{n}{n-k} \ln k\})$ for $n < k(2k-1)$ [23].

Thus for undirected MPk-CS the following ratios follow: $3k$ for any k, $k + 2\lceil (k+1)/2 \rceil$ for $2 \leq k \leq 7$, and $O(\log^4 n)$ unless $k = n - o(n)$. Improvements over the above bounds are known only for $k \leq 2$. Calinescu and Wan [6] gave for $k = 2$ a 4-approximation algorithm for undirected MPk-CS. They also gave a $2k$-approximation algorithm for undirected MPk-ECS for arbitrary k.

For results on directed min-power connectivity problems see [5] and [26].

1.5 Our Contribution

The previous best approximation ratio for MPEMC was $\min\{r_{\max}+1, O(\log^4 n)\}$ [17]. We prove:

Theorem 1.2 *The undirected* MPEMC *admits an $O(\log n)$-approximation algorithm.*

The previously best known approximation ratio for MPk-CS was $O(\alpha + \log^4 n)$ [17], where α is the best ratio for MCk-CS. From Theorems 1.2 and 1.1 we get:

Theorem 1.3 *Undirected* MPk-CS *admits an $O(\alpha + \log n)$-approximation algorithm, where α is the best ratio for* MCk-CS.

For the special case in which the cost of edges satisfy (weak) triangle inequalities, we can design an $O(\log n)$-approximation for MPk-CS. This setting makes sense for geometric graphs that is justified in the the setting of wireless networks[1]. In this algorithm, for the second step, we use a constant-factor approximation algorithm for metric MCk-CS by Khuller and Ragavachari [21]. This

[1]See [16] for details of the weak triangle inequality and the motivation in wireless networks.

proof can be done by proving the power variant of Lemma 4.6 in [21]. The description and analysis of this algorithm is left to the full version of the paper.

Remark It is rare that min-power and min-cost problems would be related with respect to approximation. However, Theorem 1.3 implies that unless MCk-CS admits a better than $O(\log n)$ approximation ratio, the min-power version MPk-CS and the min-cost version MCk-CS of the k-connected subgraph problem are equivalent. Currently, the best known ratio for the min-cost version is $O(\ln^2 k \cdot \min\{\frac{n}{n-k}, \frac{\sqrt{k}}{\ln k}\})$ for both directed and undirected graphs [23], and $O(\ln k)$ for undirected graphs with $n \geq 2k^2$ [8].

We also consider the case of small requirements which often arise in practical networks. For MPEMC with $0,1$-requirements the previously best known ratio was 2 [17]. We prove:

Theorem 1.4 MPEMC *with* $0,1$*-requirements admits a* $3/2$*-approximation algorithm.*

Theorem 1.5 *Undirected* MPk-ECS *with* k *arbitrary and undirected* MPk-CS *with* $k \in \{2,3\}$ *admit a* $(2k-1/3)$*-approximation algorithm.*

For $k = 2$, Theorem 1.5 improves the best previously known ratio of 4 [6] to $3\frac{2}{3}$. For $k = 3$ the improvement is from 7 to $5\frac{2}{3}$.

Finally, we consider the MPSN problem. Williamson et. al. [28] gave a $2r_{\max}$-approximation algorithm for the Min-Cost case, and then this was improved to $2H(r_{\max})$ in [15]. The currently best known approximation ratio for the min-cost case 2 [18]. We show that the algorithm of [28, 15] for the min-cost case, has approximation ratio $4r_{\max}$ for the min-power variant MPSN.

Theorem 1.6 *Undirected* MPSN *admits a* $4r_{\max}$*-approximation algorithm.*

To illustrate the performance of our algorithms in practical applications, we perform some experiments by implementing our algorithm for k-connectivity and show large improvements for randomly generated networks compared to some known algorithms. This is summarized in Appendix C.

Theorem 1.2, 1.4, 1.5, and 1.6 are proved in Sections 2, 3, 4, and Appendix A, respectively.

1.6 Min-power versus min-cost: some examples

Note that $p(G)$ differs from the ordinary cost $c(G) = \sum_{e\in E} c(e)$ of G even for unit costs; for unit costs, if G is undirected, then $c(G) = |E|$ and (if G has no isolated nodes) $p(G) = |V|$. For example, if E is a perfect matching on V then $p(G) = 2c(G)$. If G is a clique then $p(G)$ is roughly $c(G)/\sqrt{|E|/2}$. For directed graphs, the ratio of the cost over the power can be equal to the maximum outdegree, e.g., for stars with unit costs. The following statement (c.f., [17]) shows that these are the extremal cases for general edge costs.

Proposition 1.7 $c(G)/\sqrt{|E|/2} \leq p(G) \leq 2c(G)$ *for any undirected graph* $G = (V,E)$*, and if* G *is a forest then* $c(G) \leq p(G) \leq 2c(G)$*. For any directed graph* G *holds:* $c(G)/\Delta(G) \leq p(G) \leq c(G)$*, where* $\Delta(G)$ *is the maximum outdegree of a node in* G*.*

Min-power problems are usually harder than their min-cost versions. The min-power spanning tree problem is APX-hard [9]. The problem of finding min-cost k pairwise edge-disjoint paths is in P (this is the min-cost k-flow problem, c.f., [10]) while the directed min-power variant is unlikely to have even a polylogarithmic approximation [17]. Another directed example is finding an arborescence rooted at s, that is a subgraph that contains an sv-path for every node v. The min-cost case is in P (c.f., [10]), while the min-power variant is at least as hard as the Set-Cover problem. However, if the paths are required to be to s, the min-power case is equivalent to the min-cost case.

For *min-cost* problems, a standard reduction to reduce the undirected variant to the directed one is: replace every undirected edge $e = uv$ by two anti-parallel directed edges uv, vu of the same cost as e, find a solution D to the directed variant and take the underlying graph G of D. However, this is not at all the case for *min-power* problems. For example, our algorithm for the undirected MPEMC uses as a subroutine a $(1-1/e)$-approximation algorithm for the max-coverage with group budget constrains problem (see [1]). However, if the graph is undirected, every added edge increases the power of both sides and so the problem seems much more complicated. Indeed, the reduction described does not work for min-power problems, e.g., for MPEMC, since the power of the underlying graph of G can be much larger than that of G, e.g., if G is a star. Hence, it may happen that an algorithm for the directed case will select only one of the two anti-parallel edges, and this does not correspond to a solution for the undirected case.

The contrast is even sharepr for PBMEMC. The directed version can easily be approximated within a constant. We give strong evidence that the undirected PBMEMC may not admit a good approximation algorithm (e.g., with a constant or a polylogarithmic approximation ratio) even for unit costs and unit weights. The *Densest k-Subgraph* problem is given a graph $\mathcal{G} = (V, \mathcal{E})$ to find a subgraph of $\mathcal{G}$ with k nodes and maximum number of edges. The best known approximation ratio for the Densest k-subgraph problem is roughly $n^{-1/3}$ [12], and in spite of numerous attempts to improve it, this ratio holds for almost 10 years.

Proposition 1.8 *If there exists a ρ-approximation algorithm for undirected* PBMEMC *with unit costs, then there exist a ρ-approximation algorithm for the Densest k-Subgraph problem.*

Proof: Given an instance $\mathcal{G} = (V, \mathcal{E})$ of the Densest k-Subgraph problem, define an instance $\mathcal{G}, r, P$ with unit costs for PBMEMC as follows: $r(v) = k - 1$ for all $v \in V$ and the power budget is $P = k$. Then the problem is to find a node subset $U \subseteq V$ with $|U| = k$ so that the number of edges in the subgraph induced by U in $\mathcal{G}$ is maximum. The later is the Densest k-Subgraph problem. □

1.7 Techniques used

In most of our algorithm, we use new methods to relate power problems on undirected graphs to special carefully chosen power problems on directed graphs. For example, our algorithm for MPEMC works in iterations: at every iteration a threshold that depends on the residual demands is chosen and edges of cost above the threshold are classified as "dangerous". We will show that among the non-dangerous edges there exists a partial solution of power $O(\mathsf{opt})$ that covers a fraction of the demands. Such a partial solution is found by solving a related directed PBMEMC instance; the later is reduced to the max-coverage with group budget constrains problem which can be approximated within $(1 - 1/e)$ [1]. This leads to the desired $O(\log n)$ approximation. As was mentioned, this approach cannot be implemented directly (without deleting the dangerous edges) as the undirected

PBMEMC is unlikely to have a good approximation, see Proposition 1.8.

For MPk-CS with $k = 2, 3$ we use a combination of techniques such as *uncrossing*, in order to prove limitations on the out-degree of some directed solutions, and the known approximations for the min-power spanning tree problem and more. We also use the fact that for $k = 2, 3$ a graph is k connected if and only if it is "k-inconnected" to some node s. We derive some results on min-power connectivity problems in directed graphs that are of independent interest.

For approximating 0, 1-MPEMC we use a technique similar to the one that is used for the Min-Power spanning Tree problem [2] of decomposing optimal solution into small parts, and thus reducing with some penalty the problem to an easier problem in hypergraphs; in our case the reduction is to the min-cost case.

2 Proof of Theorem 1.2

We show an $O(\log n)$-approximation algorithm for (undirected) *bipartite* MPEMC where $\mathcal{G} = (A + B, \mathcal{E})$ is bipartite and $r(a) = 0$ for every $a \in A$.

Lemma 2.1 *If there exists a ρ-approximation algorithm for bipartite* MPEMC *then there exists a 2ρ-approximation algorithm for general* MPEMC.

Proof: Given an instance $\mathcal{G} = (V, \mathcal{E}), c, r$ of MPEMC, construct an instance $\mathcal{G}' = (V' = A + B, \mathcal{E}'), c', r'$ of bipartite MPEMC as follows. Let $A = \{a_v : v \in V\}$ and $B = \{b_v : v \in V\}$ (so each of A, B is a copy of V) and for every $uv \in \mathcal{E}$ add two edges: $a_u a_v$ and $a_v a_u$ each with cost $c(uv)$. Also, set $r'(b_v) = r(v)$ for every $b_v \in B$ and $r'(a_v) = 0$ for every $a_v \in A$. Given $F' \subseteq \mathcal{E}'$ let $F = \{uv \in \mathcal{E} : a_u b_v \in F' \text{ or } a_v b_u \in F'\}$ be the edge set in $\mathcal{E}$ that corresponds to F'. Now compute an r'-edge cover E' in $\mathcal{G}'$ using the ρ-approximation algorithm and output the edge set $E \subseteq \mathcal{E}$ that corresponds to E', namely $E = \{uv \in \mathcal{E} : a_u b_v \in E' \text{ or } a_v b_u \in E'\}$.

It is easy to see that if F' is an r'-edge cover then F is an r-edge cover. Furthermore, if for every edge in F correspond two edges in F' ($|F'| = 2|F|$), then F is an r-edge cover if, and only if, F' is an r'-edge cover. The later implies that $\mathsf{opt}' \leq 2\mathsf{opt}$, where opt and opt' is the optimal solution value to $\mathcal{G}, c, r$ and $\mathcal{G}', c', r'$, respectively. Consequently, E is an r-edge cover, and $p_E(V) \leq p_{E'}(V') \leq \rho\mathsf{opt}' \leq 2\rho\mathsf{opt}$. □

We henceforth prove that bipartite MPEMC admits an $O(\log n)$-approximation algorithm. The *residual requirement* of $v \in V$ w.r.t. an edge set I is

$$r_I(v) = \max\{r(v) - d_I(v), 0\} .$$

The approximation is performed by approximating in iterations the directed budgeted version PBMEMC of MPEMC. Before we describe the reduction of undirected MPEMC to directed PBMEMC, we explain how to efficiently approximate directed PBMEMC. We reduce directed PBMEMC to the max-coverage with group budget constrains problem that can be approximated within $(1 - 1/e)$ [1].

Instance: A bipartite graph $\mathcal{G} = (A + B, \mathcal{E})$, costs $\{c(a) : a \in A\}$, requirements $\{r(b) : b \in B\}$, budget P, and a partition $\mathcal{A}$ of A.

Objective: Find $S \subseteq A$ with $c(S) \leq P$ and $|S \cap A^i| \leq 1$ for every $A^i \in \mathcal{A}$ and $\sum_{b \in B} \min\{d(S,b), r(b)\}$ maximum.

Given an instance of directed PBMEMC build a bipartite instance $\mathcal{G}$ for the max-coverage with group budget constrains as follows. The set A contains a node a_i^v for every $v \in V$ and $e_i \in \delta_{\mathcal{E}}(v)$. The set B contains a copy v' for every $v \in V$. The vertex a_i^v is given cost $c_i = c(e_i)$. The set $A_v = \{a_1^v, a_2^v, \ldots\}$ is declared a group in the partition $\mathcal{A}$ of A. Now, add an edge (a_i^v, u') into $\mathcal{G}$ if and only if there is $e' = (v,u) \in \delta_{\mathcal{E}}(v)$ and $c(e') \leq c(e_i)$. Intuitively, a_i^v represents the choice of e_i as the largest cost edge of v in the solution. Namely, the choice of power $p(v) = c(e_i^v)$ for v. Hence, the nodes joined to v with edge-costs at most $c(e_i)$ can be covered by a_i^v obeying this power choice.

Clearly, if we are given a max-coverage solution for $\mathcal{G}$ obeying the budget P so that at most one vertex is chosen out of every group A_v, this solution defines in a unique way a solution to directed PBMEMC of the same power, and vice versa. Therefore, directed PBMEMC admits a $(1-1/e)$-approximation algorithm by [1].

The main challange remaning is to transform undirected MPEMC into directed PBMEMC without incurring a large loss in the cost. This is described now. The ultimate goal is to prove:

Lemma 2.2 *For bipartite* MPEMC *there exists a polynomial time algorithm that given an integer τ and $\gamma > 1$ either establishes that $\tau <$ opt or returns an edge set $I \subseteq \mathcal{E}$ such that*

$$p_I(V) \leq (\gamma+1)\tau \tag{1}$$

$$r_I(B) \leq (1-\beta)r(B) \ , \tag{2}$$

where $\beta = (1-1/e)(1-1/\gamma)$.

Note that if $\tau <$ opt the algorithm may return a edge set I that satisfies (1) and (2); if the algorithm declares "$\tau <$ opt" then this is correct. An $O(\log n)$-approximation algorithm for the bipartite MPEMC easily follows from Lemma 2.2:

While $r(B) > 0$ do
  Find the least integer τ so that the algorithm in Lemma 2.2 returns an edge set I
  so that (1) and (2) holds.
  $E \leftarrow E + I$, $\mathcal{E} \leftarrow \mathcal{E} - I$, $r \leftarrow r_I$.
End While

We note that the least integer τ as in the main loop can be found in polynomial time using binary search. For any constant $\gamma > 1$, say $\gamma = 2$, the number of iterations is $O(\log r(B))$, and at every iteration an edge set of power at most $(1+\gamma)$opt is added. Thus the algorithm can be implemented to run in polynomial time, and has approximation ratio $O(\log r(B)) = O(\log(n^2)) = O(\log n)$.

In the rest of this section we prove Lemma 2.2. Let τ be an integer and let $R = r(B) = \sum_{b \in B} r(b)$. An edge $ab \in \mathcal{E}$ with $b \in B$ is *dangerous* if $c(ab) \geq \gamma\tau \cdot r(b)/R$. Let $\mathcal{I}$ be the set of non-dangerous edges in $\mathcal{E}$.

Lemma 2.3 *Assume that $\tau \geq opt$. Let F be a set of dangerous edges with $p_F(B) \leq \tau$. Then $r_F(B) \geq R(1-1/\gamma)$. Thus $r_{\mathcal{I}}(B) \leq R/\gamma$.*

Proof: Let $D = \{b \in B : d_F(b) > 0\}$. We show that $r(D) \leq R/\gamma$, implying $r_F(V) \geq R - r(D) \geq R(1 - 1/\gamma)$. Since all the edges in F are dangerous, $p_F(b) \geq \gamma\tau \cdot r(b)/R$ for every $b \in D$. Thus

$$\tau \geq opt \geq \sum_{b \in D} p_F(b) \geq \sum_{b \in D} (\gamma\tau \cdot r(b)/R) = \frac{\gamma\tau}{R} \sum_{b \in D} r(b) = \frac{\gamma\tau}{R} r(D) \ .$$

For the second statement, note that there exists $E \subseteq \mathcal{E}$ with $p_E(V) \leq \tau$ so that $r_E(B) = 0$. Thus $r_I(B) \leq R/\gamma$ holds for the set I of non-dangerous edges in E. As $I \subseteq \mathcal{I}$, the statement follows. □

Lemma 2.4 $p_{\mathcal{I}}(B) \leq \gamma\tau$.

Proof: Note that $p_{\mathcal{I}}(b) \leq \gamma\tau \cdot r(b)/R$ for every $b \in B$. Thus:

$$p_{\mathcal{I}}(B) = \sum_{b \in B} p_{\mathcal{I}}(b) \leq \sum_{b \in B} (\gamma\tau \cdot r(b)/R) = \frac{\gamma\tau}{R} \sum_{b \in B} r(b) = \gamma\tau \ .$$

□

Lemmas 2.3 and 2.4 imply that we may ignore the dangerous edges and still be able to cover a fraction of the demand. Once dangerous edges are ignored, the algorithm does not need to take the power incured in B into account. Even the choice of all the non-dangerous edges will only incurr an $O(\mathsf{opt})$ cost on the B side. Therefore, the problem was transformed into *directed* PBMEMC. The algorithm is as follows:

1. With budget τ, compute an edge set $I \subseteq \mathcal{I}$ using the $(1 - 1/e)$-approximation algorithm for directed PBMEMC.
2. *If* $r_I(B) \leq (1 - \beta)R$ (recall that $\beta = (1 - 1/e)(1 - 1/\gamma)$) then output I;
   Else declare "$\tau < \mathsf{opt}$".

We show that if $\tau \geq \mathsf{opt}$ then the algorithm outputs an edge set I that satisfies (1) and (2). By Lemma 2.3, if the algorithm returns an edge set I then (1) holds for I, and if the algorithm declares "$\tau < \mathsf{opt}$" then this is correct. All the edges in I are not dangerous, thus $p_I(B) \leq \gamma\tau$ by Lemma 2.4. As we used budget τ, $p_I(A) \leq \tau$. Thus $p_I(V) = p_I(A) + p_I(B) \leq (1 + \gamma)\tau$.

3 Proof of Theorem 1.4

Given $S \subseteq V$ we say that an edge set F on V is an S-*cover*, if every node in S has an edge in F incident to it. Note that $0,1$-MPEMC is equivalent to the Min-Power S-Cover problem, where $S = \{v \in V : r(v) = 1\}$. Our approach for Min-Power S-Cover is inspired by the "decomposition method" used in [2] for the Min-Power Spanning Tree problem: a reduction to the min-cost case in 3-uniform hypergraphs with loss of 5/3 in the approximation ratio. We reduce Min-Power S-cover to Min-Cost S-Cover in 2-uniform hypergraphs (that is, in graphs, where the problem is solvable in polynomial time, c.f., [10]) with loss of 3/2 in the approximation ratio. That is, given an instance $\mathcal{G}, S$ of Min-Power S-Cover, we construct in polynomial time an instance $\mathcal{G}', S$ of Min-Cost S-Cover such that $\mathsf{opt}(\mathcal{G}') \leq 3\mathsf{opt}(\mathcal{G})/2$ and such that for any feasible solution F' to $\mathcal{G}'$ corresponds a feasible solution F to $\mathcal{G}$ with $p(F) \leq c'(F')$.

Clearly, any inclusion minimal S-cover is a union of node disjoint stars. Let F be (an edge set of) a star with center v_0. A partition $\mathcal{F} = \{F_1, \ldots, F_{\ell+1}\}$ of F into stars is a t-*decomposition* of F if $|F_{\ell+1}| \leq t - 1$ and any other part has at most t edges; $F_{\ell+1}$ covers all the nodes its edges are incident to (in particular, it covers v_0) while each part in $\mathcal{F} - F_{\ell+1}$ covers the nodes its edges are incident to except v_0 (so every node is covered exactly once). The power $p(\mathcal{F}) = \sum_{F_j \in \mathcal{F}} p(F_j)$ of $\mathcal{F}$ is the sum of the powers of its parts. A t-decomposition of a collection of stars is defined similarly.

Lemma 3.1 *Any star F with costs c has a t-decomposition $\mathcal{F}$ so that $p(\mathcal{F}) \leq (1+1/t)p(F)$.*

Proof: Let v_0 be the center of F, let $\{v_1, \ldots, v_d\}$ be the leaves of F, and let $e_i = v_0 v_i$ and $c_i = c(e_i)$ for $i = 1, \ldots, d$. Assume w.l.o.g. that $c_1 \geq c_2 \geq \cdots \geq c_d \geq 1$. Define a t-decomposition $\mathcal{F}$ of F as follows. Let $\ell = \lfloor (d-1)/t \rfloor$, and set: $F_j = \{e_{(j-1)t+1}, \ldots, e_{jt}\}$ for $j = 1, \ldots \ell - 1$ and $F_\ell = \{e_{(\ell-1)t+1}, \ldots, e_d\}$. Note that $p(F) = c(F) + c_1$ and $c_{(j-1)t+1} \leq c(F_j)/t$ for $j = 2, \ldots, \ell$; the later is since $e_{(j-1)t+1} \in F_j$, while every edge in F_{j-1} has cost larger than any edge in F_j. Therefore,

$$p(\mathcal{F}) = c(F) + c_1 + \sum_{j=2}^{\ell} c_{(j-1)t+1} \leq c(F) + c_1 + \sum_{j=2}^{\ell} c(F_{j-1})/t \leq (1+1/t)(c(F)+c_1) = (1+1/t)p(F) .$$

□

Given an instance $(\mathcal{G} = (V, \mathcal{E}; c), S)$ of MPEMC we construct an instance $(\mathcal{G}' = (S, \mathcal{E}'; c'), S)$ of min-cost edge-cover as follows. $\mathcal{G}'$ is a complete graph on S, and for every pair $u, v \in S$ let $c'(uv) = p(F_{uv})$, where F_{uv} is some minimum power $\{u, v\}$-cover that consists of one edge or two adjacent edges. Clearly, we can construct $(\mathcal{G}', S)$ and compute a min-cost S-cover in $\mathcal{G}'$ in polynomial time. The following statement that follows from Lemma 3.1 with $t = 2$ finishes the proof of Theorem 1.4.

Corollary 3.2 *If F' is a min-cost S-cover in $\mathcal{G}'$ then $F = \cup\{F_{uv} : uv \in E'\}$ is an S-cover in $\mathcal{G}$ and $p(F) \leq c'(F') \leq 3\mathsf{opt}/2$.*

4 Proof of Theorem 1.5

A (possibly directed) graph is *k-inconnected to s* if it contains k internally-disjoint vs-paths for every $v \in V$. When the paths are only required to be edge-disjoint the graph is *k-edge-inconnected to s*. Note that a graph is k-connected (resp., *k-edge-connected*) if it is k-inconnected (resp., k-edge-inconnected) to every $s \in V$. We need to consider the following "augmentation" version of the problem, where $\mathcal{G}$ contains a subgraph $G_0 = (V, E_0)$ of power zero which is k_0-inconnected to s. The goal is to augment G_0 by a min-power edge set $F \subseteq \mathcal{I} = \mathcal{E} - E_0$ so that the resulting graph $G = G_0 + F$ is k-inconnected to s. That is:

Min-Power (k_0, k)-Inconnectivity Augmentation (MP(k_0,k)-IA):
Instance: A k_0-inconnected to s graph $G_0 = (V, E_0)$, an edge set $\mathcal{I}$ on V, cost function c on $\mathcal{I}$, and an integer $k > k_0$.
Objective: Find a min-power edge set $F \subseteq I$ so that $G = G_0 + F$ is k-inconnected to s.

When G is required to be k-edge-inconnected to s we get the **Min-Power (k_0, k)-Edge-Inconnectivity Augmentation** (MP(k_0,k)-EIA) problem. The following statement was implicitly proved in [26]; for completeness of exposition its proof is given in Appendix B.

Lemma 4.1 *If F is an inclusion minimal solution to directed* MP(k_0, k_0+1)-IA *or to directed* MP(k_0, k_0+1)-EIA*, then $d_F(v) \leq 1$ for every $v \in V$, and thus the power of F equals its cost.*

By Lemma 4.1, the augmentation problem of increasing the inconnectivity (or edge-inconnectivity) of a directed graph by 1, the min-power case is equivalent to the min-cost case; the later is solvable in polynomial time [14, 13]. As we will show later, Lemma 4.1 implies the following statement:

Lemma 4.2 *Undirected* $\mathsf{MP}(k_0, k_0+1)$-IA *and* $\mathsf{MP}(k_0, k_0+1)$-EIA *admit a* 2*-approximation algorithm.*

Theorem 1.5 easily follows by combining Lemma 4.2 with the 5/3-approximation algorithm of [2] for the Min-Power Spanning Tree problem. Indeed, we can apply the algorithm as in Lemma 4.2 sequentially to produce edge sets $F_1, \ldots, F_k$ so that $G_\ell = F_1 + \cdots + F_\ell$ is ℓ-inconnected (resp., ℓ-edge-inconnected) to s, and $p(F_1) \leq 5\mathsf{opt}/3$ (F_1 is a spanning tree computed by the 5/3-approximation algorithm of [2]) and $p(F_\ell) \leq 2\mathsf{opt}$ for $\ell = 2, \ldots, k$. Consequently, if $E = F_1 + \cdots + F_k$ then $G = (V, E)$ is k-inconnected (resp., k-edge-inconnected) to s, and

$$p(E) \leq p(F_1) + \sum_{\ell=2}^{k} p(F_\ell) \leq \frac{5}{3}\mathsf{opt} + \sum_{\ell=2}^{k} 2\mathsf{opt} = 2(k - 1/3)\mathsf{opt} \ .$$

Finally, the $(2k - 1/3)$-approximation algorithm for $\mathsf{MP}k$-CS with $k \in \{2, 3\}$ follows from the following two facts (c.f., [3]):

(i) Any undirected minimally k-connected graph has at least $|V|/3$ nodes of degree k;

(ii) For $k \in \{2, 3\}$, if s is a node of degree k in an undirected graph G, then G is k-inconnected to s if, and only if, G is k-connected.

Hence for $k \in \{2, 3\}$ undirected $\mathsf{MP}k$-CS is equivalent (via an approximation ratio preserving reduction) to the problem of finding a min-power k-inconnected to s subgraph so that the degree of s is exactly k. The later problem admits a $(2k - 1/3)$-approximation algorithm for any constant k, by trying $O(n^{k+1})$ possible choices of s and the k edges incident to it. In fact, using penalty methods (see [11]) the exhaustive search can be reduced to $O(n^2)$ possible choices of s and one edge incident to it (details omitted). This gives a $(2k - 1/3)$-approximation algorithm for $\mathsf{MP}k$-CS with $k \in \{2, 3\}$.

In the rest of this section we prove Lemma 4.2. A *biderection* of an undirected network G is a directed network obtained by replacing every edge $e = uv$ of G by two opposite directed edges uv, vu each having the same cost as e; if D is a subgraph of a bidirection of G, then we say that G is the *underlying network (or graph)* of D. Clearly, if D is a bidirection of G then $p(G) = p(D)$.

Lemma 4.3 $p(G) \leq (\Delta(D) + 1)p(D)$ *for the underlying graph* G *of a directed network* D.

Proof: By induction on the number m of edges in D. For $m = 1$ the statement is obvious. Assume that the statement is true for digraphs with at most $m - 1$ edges. Let $v \in V$ be a node in D of maximum power $c_{\max}$. Let $D' = D - \delta_D(v)$ and let G' be the underlying graph of D'. Clearly, $p(D') = p(D) - c_{\max}$ and $p(G') \geq p(G) - (\Delta(D)+1)c_{\max}$. Combining with the induction hypothesis gives: $p(G) \leq p(G') + (\Delta(D) + 1)c_{\max} \leq (\Delta(D) + 1)(p(D) + c_{\max}) = (\Delta(D) + 1)p(D)$. □

The 2-approximation algorithm for $\mathsf{MP}(k_0, k_0 + 1)$-$\mathsf{IA}$ is as follows:

1. Let D_0 and $\mathcal{A}$ be the bidirections of G_0 and $\mathcal{I}$, respectively.
2. Compute an optimal edge set $A \subseteq \mathcal{A}$ so that $D_0 + A$ is $(k_0 + 1)$-inconnected to s.
3. Output the underlying edge set I of A.

Let I^* be an optimal solution to $\mathsf{MP}(k_0, k_0+1)$-IA instance (so $p(I^*) = \mathsf{opt}$) and let A^* be the bidirection of I^*. Note that $\Delta(A) \leq 1$, by Lemma 4.1. This implies $p(I) \leq (\Delta(A)+1)p(A) \leq 2p(A)$, by Lemma 4.3. Thus we have:

$$p(I) \leq 2p(A) \leq 2p(A^*) = 2p(I^*) = 2\mathsf{opt} \; .$$

The algorithm and the analysis for ($\mathsf{MP}(k_0, k_0+1)$-EIA) is similar. This finishes the proof of Lemma 4.2, and thus the proof of Theorem 1.5 is complete.

References

[1] A. Ageev and M. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Comb. Optim*, 8(3):307–328, 2004.

[2] E. Althaus, G. Calinescu, I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment in ad-hoc wireless networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1889–1894, 2003.

[3] V. Auletta, Y. Dinitz, Z. Nutov, and D. Parente. A 2-approximation algorithm for finding an optimum 3-vertex-connected spanning subgraph. *Journal of Algorithms*, 32(1):21–30, 1999.

[4] M. Bahramgiri, M. Hajiaghayi, and V. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms wireless multi-hop networks. In *Proceedings of the 11th IEEE International Conference on Computer Communications and Networks (ICCCN)*, pages 392–398. IEEE Press, 2002.

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

[6] G. Calinescu and P. J. Wan. Range assignment for high connectivity in wireless ad hoc network. In *Adhoc-Now*, pages 235–246, 2003.

[7] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. In *Proceedings of fiftieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. 2003. To appear.

[8] J. Cheriyan, S. Vempala, and A. Vetta. An approximation algorithm for the minimum-cost k-vertex connected subgraph. *SIAM Journal on Computing*, 32(4):1050–1055, 2003.

[9] A. E. F. Clementi, P. Penna, and R. Silvestri. Hardness results for the power range assignmet problem in packet radio networks. In *Workshop on Approximation algorithms (APPROX)*, pages 197–208, 1999.

[10] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Scrijver. *Combinatorial Optimization.* Wiley, 1998.

[11] Y. Dinitz and Z. Nutov. A 2-approximation algorithm for finding an optimum 4,5-vertex-connected spanning subgraph. *Journal of Algorithms*, 32(1):31–40, 1999.

[12] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, (29):410–421, 2001.

[13] A. Frank. Increasing the rooted-connectivity of a digraph by one. *Mathematical Programming*, 84(3):565–576, 1999.

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

[15] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. B.Shmoys, E. Tardos, and D. P. Williamson. Improved approximation algorithms for network design problems. In *Symposium on Discrete Algorithms (SODA)*, pages 223–232, 1994.

[16] M. T. Hajiaghayi, N. Immorlica, and V. S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking (MOBICOM)*, pages 300–312, 2003.

[17] M. T. Hajiaghayi, G. Kortsarz, V. S. Mirrokni, and Z. Nutov. Power optimization for connectivity problems. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 349–361, 2005.

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

[19] S. Khuller. *Approximation algorithms for for finding highly connected subgraphs,* Chapter 6 in *Approximation Algorithms for NP-hard problems,* D. S. Hochbaum Ed., pages 236-265. PWS, 1995.

[20] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[21] S. Khuller and B. Raghavachari. Improved approximation algorithms for uniform connectivity problems. *Journal of Algorithms*, 21:434–450, 1996.

[22] G. Kortsarz and Z. Nutov. Approximating node-connectivity problems via set covers. *Algorithmica*, 37:75–92, 2003.

[23] G. Kortsarz and Z. Nutov. Approximating k-node connected subgraphs via critical graphs. *SIAM J. Comput.*, 35(1):247–257, 2005.

[24] G. Kortsarz and Z. Nutov. Approximating minimum cost connectivity problems, in *Approximation Algorithms and Metaheuristics,* T. F. Gonzalez ed.,. 2005. To appear.

[25] L. Li, J. Halpern, V. Bahl, Y. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *Proceedings of ACM Symposium on Principle of Distributed Computing (PODC)*, pages 264–273. 2001.

[26] Z. Nutov. Approximating minimum power covers of intersecting families and directed connectivity problems. *Workshop on Approximation algorithms (APPROX)*, 2006. To appear.

[27] R. Wattenhofer, L. Li, V. Bahl, and Y. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proceedings of twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1388–1397. IEEE Press, 2001.

[28] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A primal-dual approximation algorithm for generalized steiner network problems. *Combinatorica*, 15:435–454, 1995.

Appendix A: Proof of Theorem 1.6

We need some definitions and a description of certain results from [28, 15].

Min-cost/power Steiner Network problem can be formulated as a *set-function edge-cover problem*. Let $h : 2^V \rightarrow Z_+$ be a *set-function* defined on a groundset V. An edge set E on V is an *h-cover*, if $d_E(X) \geq h(X)$ for every $X \subseteq V$. For Steiner Network problems, an appropriate choice of h is as follows. By Menger's Theorem, E is a feasible solution to min-cost/power Steiner network problem if, and only if, $d_E(X) \geq R(X)$ for all $\emptyset \subset X \subset V$, where $R(X) = \max\{r(u, v) : u \in X, v \in V - X\}$ (and $R(\emptyset) = R(V) = 0$). That is

$$d_E(X) \geq h(X) \equiv \max\{0, R(X)\} \quad \forall\, \emptyset \subseteq X \subseteq V. \tag{3}$$

The function h defined above is *skew-supermodular*, that is $h(\emptyset) = 0$ and for every $X, Y \subseteq V$ with $h(X) > 0$, $h(Y) > 0$ at least one of the following holds:

$$h(X) + h(Y) \leq h(X \cap Y) + h(X \cup Y) \tag{4}$$

$$h(X) + h(Y) \leq h(X - Y) + h(Y - X) \tag{5}$$

Note that h is also *symmetric*, that is, $h(X) = h(V - X)$ for all $X \subseteq V$.

Several connectivity problems can be formulated as (min-cost/power) edge cover problems of a skew-supermodular function, see [24]. A seminal paper of Jain [18] gives a 2-approximation algorithm for finding a min-cost edge-cover of an arbitrary skew-supermodular set function h, provided certain queries related to h can be answered in polynomial time (note that h is usually not given explicitly). For h defined in (3) these queries can be realized via max-flows, which implies a 2-approximation algorithm for the min-cost Steiner network problem. Earlier, Williamson et. al [28] gave an algorithm with approximation ratio $2h_{\max}$, which was improved later to $2H(h_{\max})$ by Goemans et. al [15].

Given a set function q, let $\hat{q}(X) = 1$ if $q(X) = q_{\max}$ and $hq(X) = 0$ otherwise, where $q_{\max} = \max_{X \subseteq V} q(X)$. It is easy to see that any inclusion minimal edge-cover of a $\{0, 1\}$-valued set function is a forest. For an edge set E, let p_E be defined as follows: $p_E(X) = \max\{p(X) - d_E(X), 0\}$. It is well known that if h is skew supermodular, so is h_E (for any edge set E), see [18]. Consider the following algorithm that applies on an arbitrary set-function h, and begins with $E = \emptyset$.

While there is $X \subseteq V$ with $h_E(X) > 0$ do:
1. Find an $\hat{h}_E$-cover $F \subseteq \mathcal{E} - E$;
2. $E \leftarrow E + F$.

End While

The approximation ratio of the algorithm depends on step 1. A set function is called *uncrossable* if it is $\{0, 1\}$-valued skew supermodular. It is easy to see that if q is skew supermodular, so is $\hat{q}$, that is $\hat{q}$ is uncrossable. Williamson et. al [28] gave an algorithm that finds an edge cover of an uncrossable function $\hat{q}$ of cost at most twice the optimum of the following LP-relaxation:

$$\min\{\sum_{e \in \mathcal{E} - E} c(e)x_e : \sum_{e \in \delta(X)} x_e \geq \hat{q}(X)\ \forall X \subseteq V, x_e \geq 0\}\ . \tag{6}$$

Williamson et. al [28] proved:

Theorem 4.4 ([28]) *For h defined by (3) the above algorithm can be implemented in polynomial time, so that at any iteration for $q = h_E$ the forest F found has cost at most twice the optimal value of (6).*

Note that the number of iterations of the algorithm is at most $h_{\max}$. Thus Theorem 4.4 implies that for the min-cost Steiner network problem the algorithm has approximation ratio $2h_{\max} \leq 2r_{\max}$. Later, Goemans et. al [15] used linear programming scaling techniques to show that the approximation ratio is in fact $2H(r_{\max})$. This scaling method does not work for the min-power variant.

We can show that for the min-power variant, the algorithm of [28] has approximation ratio $4r_{\max}$. This follows from Theorem 4.4 and the second part of Proposition 1.7. Indeed, the algorithm of [28] constructs the solution from at most $r_{\max}$ forests, where each forest has cost at most $2\mathsf{opt_c}$, where $\mathsf{opt_c}$ is the optimal solution value to the min-cost variant. By Proposition 1.7, each forest has power at most $2 \cdot 2\mathsf{opt_p} = 4\mathsf{opt_p}$, where $\mathsf{opt_p}$ is the otimal solution value to the min-power variant. This completes the proof of Theorem 1.6.

Appendix B: Proof of Lemma 4.1

It would be convenient to prove Lemma 4.1 for the following problem that generalizes both $\mathsf{MP}(k_0, k_0+1)$-IA and $\mathsf{MP}(k_0, k_0+1)$-EIA. A graph $G = (V, E)$ is *ℓ-edge-connected from U to s* if there are ℓ edge-disjoint us-paths for every $u \in U$.

Directed Min-Power (U, s)-Connectivity-Augmentation
Instance: A graph $G_0 = (V, E_0)$ which is k_0-edge-connected from U to s and an edge set $\mathcal{I}$ on V with costs $\{c_e : e \in \mathcal{I}\}$ so that *every edge in $\mathcal{I}$ has its tail in U.*
Objective: Find a min-power $I \subseteq \mathcal{I}$ so that $G_0 + I$ is (k_0+1)-edge-connected from U to s.

$\mathsf{MP}(k_0, k_0+1)$-EIA is a special case of this problem when $U = V$. For $\mathsf{MP}(k_0, k_0+1)$-IA apply the following well known reduction, c.f., [13]. Given an instance $G_0 = (V, E_0), k_0, s, \mathcal{I}$ for $\mathsf{MP}(k_0, k_0+1)$-IA obtain an instance $G'_0 = (V', E'_0), U', k_0, s', \mathcal{I}', c'$ for the above problem as follows. Replace every node $v \in V$ by the two nodes v_t, v_h connected by the edge $v_t v_h$ of cost zero and replace every edge $uv \in E_0 \cup \mathcal{I}$ by the edge $u_h v_t$ having the same cost as uv (which is zero if $uv \in E_0$). Let $s' = s_t$, $U' = \{v_h : v \in V\}$, $E'_0 = \{u_h v_t : uv \in E_0\} + \{v_t v_h : v \in V\}$, and $\mathcal{I}'_0 = \{u_h v_t : uv \in \mathcal{I}\}$. This establishes a bijective correspondence between edges in $\mathcal{I}$ and the edges in $\mathcal{I}'$. It is not hard to verify (c.f., [13]) that $G'_0 = (V', E'_0)$ is k_0-edge-connected from U' to s', and that if $I' \subseteq \mathcal{I}$ corresponds to $I \subseteq \mathcal{I}$ then:
(i) I is a feasible solution if, and only if, I' is a feasible solution;
(ii) $d_I(v) = d_{I'}(v_h)$ and $d_{I'}(v_t) = 0$ for every $v \in V$ (thus I and I' have the same power).

Thus Lemma 4.1 will be proved if we show that $d_F(v) \leq 1$ for any inclusion minimal solution F to Directed Min-Power (U, s)-Connectivity-Augmentation. We say that an edge set F *covers* a set family $\mathcal{F}$ if for every $X \in \mathcal{F}$ there is an edge in F leaving X. A set family $\mathcal{F}$ on V is intersecting if $X \cap Y, X \cup Y \in \mathcal{F}$ for any intersecting $X, Y \in \mathcal{F}$. We say that $X \subseteq V - s$ is *tight* in G_0 if $X \cap U \neq \emptyset$ and $d_{G_0}(X) = k_0$. From Menger's Theorem we have:

Fact 4.5 *Let $G_0 = (V, E_0)$ be k_0-edge-connected from U to s. Then $G = G_0 + F$ is (k_0+1)-connected from U to s if, and only if, F covers all the tight sets.*

By Fact 4.5 F is an inclusion minimal solution to Directed Min-Power (U, s)-Connectivity-Augmentation if, and only if, F is an inclusion minimal cover of the family of tight sets of G_0. However, since only edges with tail in U can be added, F covers the tight sets of G_0 if, and only if, F covers the family:

$$\mathcal{F} = \{X \cap U : X \text{ is tight in } G_0\} \ . \tag{7}$$

It is well known and easy to show that (c.f. [13]):

Fact 4.6 *The family $\mathcal{F}$ defined in (7) is intersecting.*

Thus the following statement finishes the proof of Lemma 4.1.

Lemma 4.7 *If F is an inclusion minimal cover of an intersecting family $\mathcal{F}$, then $d_F(v) \leq 1$ for every $v \in V$, and thus the power of F equals its cost.*

Proof: By the minimality of F, for every $e \in F$ there exists $W_e \in \mathcal{F}$ such that $\delta_F(W_e) = \{e\}$; we call such W_e a *witness set for* e; note that e might have several distinct witness sets.

Let W_e, W_f be intersecting witness sets of two distinct edges $e, f \in F$. We claim that then $W_e \cap W_f$ is a witness for one of e, f and $W_e \cup W_f$ is a witness for the other. This implies that there cannot be $v \in V$ with $e, f \in \delta_F(v)$, as otherwise $e, f \in \delta(W_e \cap W_f)$ which is a contradiction. Thus by Proposition 1.7, the power of F equals its cost.

We now prove that $W_e \cap W_f$ is a witness for one of e, f and $W_e \cup W_f$ is a witness for the other. Note that there is an edge in F leaving $W_e \cap W_f$ and there is an edge in F leaving $W_e \cup W_f$; this is since $W_e, W_f \in \mathcal{F}$ implies that $W_e \cap W_f, W_e \cup W_f$ belong to $\mathcal{F}$ and thus each of them is covered by some edge in F. However, if for arbitrary sets X, Y an edge covers one of $X \cap Y, X \cup Y$ then it also covers one of X, Y, and if some edge covers both $X \cap Y$ and $X \cup Y$ then it must cover both X and Y. Thus no edge in $E - \{e, f\}$ can cover $W_e \cap W_f$ or $W_e \cup W_f$, so one of e, f covers $W_e \cap W_f$, and thus the other must cover $W_e \cup W_f$. □

Appendix C: Performance Evaluation

In the previous sections, we proved a worst-case bound for the performance of our algorithms compared to the optimal solution. In this section, we report our observations on the implementation of the algorithm for MPk-CS. In order to understand the effectiveness of our algorithm, we compare its output to a previous heuristic, namely the Cone-based topology control heuristic of Wattenhofer et al. [27] and Li et al. [25] and Bahramgiri et al. [4]. In this heuristic, each node increases transmission power until the angle between any pair of adjacent neighbors is at most $\frac{2\pi}{3k}$. Bahramgiri, Hajiaghayi, and Mirrokni [4] proved that if the original graph is k-connected, the resulting graph after this heuristic is also k-connected. This algorithm has an advantage of being localized; however we show that the power consumption of the resulting solution can be much worse than our algorithm based on approximating MPEMC.

We generate random networks, each with at most 50 nodes. The maximum possible power at each node is fixed at $E_{\max} = (250)^2$. We assume that the power attenuatin exponent $c = 2$. This

		CBTC Heuristic			Algorithm based on MPEMC		
Connectivity #		2	3	4	2	3	4
Density	Degree	ERR					
17	33.12	76.15	92.66	98.02	44.32	58.01	64.07
20	42.76	61.19	83.60	94.73	28.16	58.85	64.62
25	49.18	61.62	83.70	93.19	29.21	35.95	40.18
30	54.56	58.82	75.12	92.43	16.32	25.52	41.90
35	59.32	54.76	75.04	90.37	30.83	39.51	44.03

Table 1: Expended Energy Ratio for 2,3, and 4-connectivity and $c = 2$

implies a maximum communication radius R of 250 meters. We evaluate the performance of our algorithms on networks of varying density. For the performance measure, we compute the average expended energy ratio (EER) of both algorithms for these random networks:

$$EER = \frac{\text{Average Power}}{E_{\max}} \times 100.$$

We assume that the MAC layer is ideal. Our sample networks are similar to the sample networks used by Wattenhofer et al. [27], Cartigny et al. [7]. Our experimental results are summarized in Table 1.

As expected, our algorithm outperforms CBTC for all networks in our experiment. Note that the worst-case approximation factor of the algorithm based on approximating MPEMC does not depend on k. As a result, we expect that the performance of this algorithm is better compared to CBTC as k increases. One can verify this fact by observing that the performance of CBTC heuristic decreases by a larger factor from 2-connectivity to 4-connectivity. For example, EER for CBTC increases from 54.76 to 90.37 for one instance and from 76.15 to 98.02 for another instance. However, for the same instances, the EER for the algorithm based on approximating MPEMC increases from 30.83 to 44.03 and from 44.32 to 64.07, respectively. This indicates the faster diminishing performance of CBTC compared to our algorithm as k increases.