Approximating Minimum-Power Degree and Connectivity Problems

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

Abstract

Power optimization is a central issue in wireless network design. Given a (possibly directed) graph with costs on the edges, the power of a node is the maximum cost of an edge leaving it, and the power of a graph is the sum of the powers of its nodes. Motivated by applications in wireless networks, we consider several fundamental undirected network design problems under the power minimization criteria. Given a graph $\mathcal{G} = (V, \mathcal{E})$ with edge costs $\{c_e : e \in \mathcal{E}\}$ and degree requirements $\{r(v) : v \in V\}$, the Minimum-Power Edge-Multi-Cover (MPEMC) problem is to find a minimum-power subgraph of $\mathcal{G}$ 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 previous ratio $O(\log^4 n)$ of [16]. This is used to derive an $O(\log n + \alpha)$-approximation algorithm for the undirected Minimum-Power k-Connected Subgraph (MPk-CS) problem, where α is the best known ratio for the min-cost variant of the problem (currently, $\alpha = O(\ln k)$ for $n \geq 2k^2$ and $\alpha = O(\ln^2 k \cdot \min\{\frac{n}{n-k}, \frac{\sqrt{k}}{\log n}\})$ otherwise). Surprisingly, it shows that the min-power and the min-cost versions of the k-Connected Subgraph problem are *equivalent* with respect to approximation, unless the min-cost variant admits an $o(\log n)$-approximation, which seems to be out of reach at the moment.

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

†Microsoft Research, mirrokni@microsoft.com

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

§IBM, Haifa tsanko@il.ibm.com

1 Introduction

1.1 Motivation and problems considered

Wireless networks are studied extensively due to their wide applications. 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 quadratically in the transmission range. Assigning power levels to the stations (nodes) determines the resulting communication network. Conversely, given a communication network, the cost required at v only depends on the furthest node that is reached directly by v. This is in contrast with wired networks, in which every pair of stations that need to communicate directly incurs a cost. We study the design of symmetric wireless networks that meet some prescribed degree or connectivity properties, and such that the total power is minimized. An important network property is fault-tolerance, which is often measured by minimum degree or node-connectivity of the network. Node-connectivity is much more central here than edge-connectivity, as it models stations failures. Such power minimization problems were vastly studied. See for example [1, 4, 15, 16, 22, 24, 9, 5] for a small sample of papers in this area. The first problem we consider is finding a low power network with specified lower bounds on node degrees. This is the power variant of the fundamental b-Matching/Edge-Multi-Cover problem, c.f., [10]. The second problem is the Min-Power k-Connected Subgraph problem which is the power variant of the classic Min-Cost k-Connected Subgraph problem. We devise approximation algorithms for these problems, improving significantly the previously best known ratios.

Definition 1.1 *Let $G = (V, E)$ be a graph with edge-costs $\{c(e) : e \in E\}$. For $v \in V$, the* power $p(v) = p_G(v)$ of v *in G (w.r.t. c) is the maximum cost of an edge in G leaving v, i.e., $p(v) = p_E(v) = \max_{vu \in E} c(vu)$. The power of the graph is the sum of the powers of its nodes.*

Unless stated otherwise, graphs are assumed to be undirected and simple. Let $G = (V, E)$ be a graph. For $X \subseteq V$, $\Gamma_E(X) = \Gamma_G(X) = \{u \in V - X : v \in X, vu \in \mathcal{E}\}$ is the set of neighbors of X, $\delta_E(X) = \delta_G(X)$ is the set of edges leaving X, and $d_E(X) = |\delta_G(X)| = |\Gamma_G(X)|$ is the degree of X in G. Let $\mathcal{G} = (V, \mathcal{E}; c)$ be a *network*, that is, $(V, \mathcal{E})$ is a graph and c is a cost function on $\mathcal{E}$. Sometimes, we write $\mathcal{G} = (V, \mathcal{E})$ and refer to $\mathcal{G}$ as a graph. Let $n = |V|$ and $m = |\mathcal{E}|$. Given a network $\mathcal{G} = (V, \mathcal{E}; c)$, we seek to find a low power *communication network*, that is, a low power subgraph $G = (V, E)$ of $\mathcal{G}$ that satisfies some property. Two such fundamental properties are: degree constraints and fault-tolerance/connectivity. In fact, these problems are related, and we use our algorithm for the former as a tool for approximating the latter.

Definition 1.2 *Given a requirement function r on V, we say that a graph $G = (V, E)$ (or that E) is an r-*edge cover *if $d_G(v) \geq r(v)$ for every $v \in V$, where $d_G(v) = d_E(v)$ is the degree of v in G.*

Finding a minimum-cost r-edge cover is a fundamental problem in combinatorial optimization, as this is essentially the b-Matching problem, c.f., [10]. The following problem is the power variant.

Minimum-Power Edge-Multi-Cover (MPEMC):
Instance: A network $\mathcal{G} = (V, \mathcal{E}; c)$ and degree requirements $\{r(v) : v \in V\}$.
Objective: Find a min-power subgraph G of $\mathcal{G}$ so that G is an r-edge cover.

We now define our connectivity problem. A graph is *k-connected* (*k-edge-connected*) if it contains k internally-disjoint (k edge-disjoint) uv-paths for all $u, v \in V$.

Minimum-Power k-Connected Subgraph (MPk-CS):
Instance: A network $\mathcal{G} = (V, \mathcal{E}; c)$, and an integer k.
Objective: Find a minimum-power k-connected spanning subgraph G of $\mathcal{G}$.

We give improved approximation algorithms for these problems. As a tool for approximating MPEMC, we consider a special case of the following problem:

Budgeted Multi-coverage with Group Constraints (BMGC)
Instance: A bipartite graph $\mathcal{G} = (A + B, \mathcal{E})$, costs $\{c(a) : a \in A\}$, a budget P, degree requirements $\{r(b) : b \in B\}$, and a partition $\mathcal{A}$ of A.
Objective: Find $S \subseteq A$ with $c(S) \leq P$ and $\mathsf{val}(S) = \sum_{b \in B} \min\{|\Gamma_{\mathcal{G}}(b) \cap S|, r(b)\}$ maximum, so that $|S \cap A^i| \leq 1$ for every $A^i \in \mathcal{A}$.

If $\mathcal{A}$ is not a partition, but just a collection of subsets of A (even of size 2), then BMGC includes the Independent Set problem even if $r(b) = 1$ for all $b \in B$. Hence assuming that $\mathcal{A}$ partitions A is essential. BMGC generalizes both the Budgeted Maximum Coverage problem (when $\mathcal{A}$ is a partition into singletons) which admits a $(1 - 1/e)$-approximation [18], and the Maximum Coverage with Group Constraints problem in which there is no global budget P and all the requirement are 1. For this special case, [7] gave a 1/2-approximation. We also mention that BMGC belongs to the class of problems that seek to maximize a non-decreasing submodular function under certain constraints. There exists a 1/2-approximation algorithm for matroid constrains [14], and there exist a $(1-1/e)$ approximation algorithm for knapsack constrains [25]. BMGC has *both* matroid *and* knapsack constrains, and we are not aware of a technique that handles both.

Studying the approximability of BMGC is beyond the scope of this paper. To get the $O(\log n)$ approximation for MPEMC, we give a $(1 - 1/e)$-approximation algorithm for the following special case.

Definition 1.3 *A* BMGC *instance has the* Star-Property *if every $A^i \in \mathcal{A}$ admits an ordering $a_1, a_2, \ldots$ so that $\Gamma_{\mathcal{G}}(a_{j-1}) \subseteq \Gamma_{\mathcal{G}}(a_j)$. Let* BMGC* *be the restriction of* BMGC *to instances with the Star-Property.*

In "set-cover" terms, in BMGC A is a collection of sets on a ground-set B, and $\mathcal{A}$ is a partition of the sets. The Star-Property requires that every group is a nested family.

1.2 Related Work

Results on MPEMC: The Minimum-Cost Edge-Multi-Cover problem is essentially the fundamental b-Matching problem, which is solvable in polynomial time, c.f., [10]. The previously best known approximation ratio for the min-power variant MPEMC was $\min\{r_{\max} + 1, O(\log^4 n)\}$ due to [16]. The directed MPEMC generalizes the classic Minimum-Cost Set-Multi-Cover problem; the latter is a special case when for every $v \in V$ all the edges leaving v have the same cost.

Results on connectivity problems: The simplest connectivity problem is when we require the network to be connected. In this case, the minimum-cost variant is just the Minimum-Cost Spanning Tree problem, while the minimum-power variant is APX-hard. A 5/3-approximation algorithm for the Minimum-Power Spanning Tree problem is given in [1]. Minimum-cost connectivity problems for arbitrary k were extensively studied, see surveys in [17] and [21]. The best known approximation ratios for the Minimum-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 [20], 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 ([16])

(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 algorithm 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 [16] 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 [2] for $k = 2, 3$, [11] for $k = 4, 5$, and [19] for $k = 6, 7$); $\alpha = k$ for $k = O(\log n)$ [19], $\alpha = 6H(k)$ for $n \geq k(2k-1)$ [8], and $\alpha = O(\ln k \cdot \min\{\sqrt{k}, \frac{n}{n-k} \ln k\})$ for $n < k(2k-1)$ [20]. 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 [5] gave a 4-approximation algorithm for the case $k = 2$ of undirected MPk-CS. They also gave a $2k$-approximation algorithm for undirected MPk-ECS for arbitrary k. For further results on other minimum-power connectivity problems, among them problems on directed graphs see [4, 16, 24, 22].

1.3 Our Results

The previous best approximation ratio for MPEMC was $\min\{r_{\max} + 1, O(\log^4 n)\}$ [16]. We prove:

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

This result uses the following statement:

Lemma 1.3 BMGC* *admits a* $(1 - 1/e)$*-approximation algorithm.*

The previously best known ratio for MPk-CS was $O(\alpha + \log^4 n)$ [16], where α is the best ratio for MCk-CS. From Theorems 1.2 and 1.1, and from [8], we get:

Theorem 1.4 MPk-CS *admits an* $O(\alpha + \log n)$*-approximation algorithm, where* α *is the best ratio for* MCk-CS*. In particular, for* $n \geq 2k^2$*,* MPk-CS *admits an* $O(\log n)$*-approximation algorithm.*

Theorem 1.4 implies that the min-cost and the min-power variants of the k-Connected Subgraph problem are equivalent with respect to approximation, unless the min-cost variant admits a better than $O(\log n)$-approximation; the latter seems to be out of reach at the moment, see [20, 8]; the best known ratio for MCk-CS when $k = n - o(n)$ is $\tilde{O}(\sqrt{n})$ [20]. This equivalence can turn useful for establishing a lower bound for MCk-CS. In particular, if we can show an approximation threshold of $\Omega(\log^{1+\varepsilon} n)$ for MPk-CS, then the same threshold applies for MCk-CS; on the other hand, if MPk-CS admits a logarithmic ratio, then so does MCk-CS. Note that for $n \geq 2k^2$, our ratio for MPk-CS is $O(\log n)$, and this matches the best known ratio for MCk-CS with $n \geq 2k^2$ of [8].

Lemma 1.3 is proved in Section 2. Theorem 1.2 is proved in Section 3.

1.4 Techniques

The technique used for approximating MPEMC is new and is not similar to the weaker approximation given in [16]. For MPk-CS we use the easy reduction from approximating MPk-CS to approximating MPEMC [16] and rely on our new MPEMC approximation. Thus, designing new approximation for MPEMC is the crux of the matter. Approximating MPEMC turned up to be a rather challenging task (some reasons for that are explained in Section 1.5). Intuitively, the difficulty is that adding an edge to the solution may cause the increase in power for both endpoints of the edge. Thus if we are given a budget and attempt to satisfy as much demand as possible within the budget, this turns out to be as hard as the dense k-subgraph problem (see [13]), as explained in Section 1.5. The algorithm of [16] is unsuited for deriving an $O(\log n)$ ratio for MPEMC, as it pays a $\log^2 n$ factor in the ratio, from the get-go. Hence, a completely new strategy is required. The ideas of our algorithm are summarized as follows:

1. **Reduction to bipartite graphs:** We reduce the problem to a bipartite graph $\mathcal{G}' = (A + B, \mathcal{E}')$, with each of A and B being a copy of V. Thus every node has two occurrence, one in A and one in B. The side of B has degree requirement while A is the "covering side" and has no demands. This reduction is simple, but it is crucial for technical reasons.

2. **Ignoring dangerous edges:** The algorithm works in iterations. At every iteration some edges are declared "dangerous", hence forbidden for use in the current iteration; *this is our main new technique.* Classifying edges as dangerous depends not only on their cost, but also on the *residual demand*; hence the set of dangerous edges changes from iteration to iteration. We prove that at any specific iteration, the contribution of dangerous edges to the cover cannot be too large, as they are too expensive to cover "too much" of the demand. Intuitively, ignoring dangerous edges is a trick that allows us to focus on minimizing the power of the nodes in A only; even if every $b \in B$ is touched by its most expensive non-dangerous edge, we are still able to appropriately bound the increase in the power of the nodes in B. We believe that this technique will have further applications.

3. **Reduction to the BMGC*:** At every iteration of the algorithm the goal is to pay $O(\mathsf{opt})$ in the power increase, and reduce the sum of the (residual) demands by a

constant fraction. Hence, after $O(\log n)$ iterations, all the requirements are satisfied, and the $O(\log n)$ ratio follows. In every iteration, after the dangerous edges are ignored, we are able to cast the problem we need to solve as an instance of BMGC*.

4. **Approximating** BMGC*: We design a simple "local-replacement" $(1-1/e)$-approximation algorithm for BMGC*. The analysis, which is quite involved, generalizes the analysis of the algorithm of [18] for the Budgeted Maximum Coverage problem. The difference is that the [18] algorithm only adds elements, and hence it is not a local replacement algorithm.

Finally, since the problem in question is of considerable practical importance, we measure the performance of our algorithms in practice. We perform some experiments by implementing our algorithm for k-connectivity and show improvements for randomly generated networks compared to some known algorithms.

1.5 Power Optimization vs. Cost Optimization: A Comparison

Theorem 1.4 implies that, unless MCk-CS admits a better than $O(\log n)$ approximation ratio, the minimum-power version MPk-CS and the minimum-cost version MCk-CS of the k-Connected Subgraph problem are *equivalent* with respect to approximation: one of the problems admits a polylogarithmic approximation if, and only if, the other does, and the same holds for superlogarithmic approximation thresholds. This near approximability equivalence of MPk-CS and MCk-CS is a rare and surprising example in power versus cost problems. Typically, problems behave completely differently in the minimum-power versus the minimum-cost models. Power problems are "threshold" type of problems, in the sense that, if many edges of the same (maximum) cost touch a node v, or just one such edge touches v, the power of v is the same.

We now compare in detail some additional aspects of power versus cost problems. 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., [16]) shows that these are the extremal cases for general edge costs.

Proposition 1.5 *$c(G)/\sqrt{|E|/2} \leq p(G) \leq 2c(G)$ 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.

Minimum-power problems are usually harder than their minimum-cost versions. The Minimum-Power Spanning Tree problem is APX-hard. The problem of finding minimum-cost k pairwise edge-disjoint paths is in P (this is the Minimum-Cost k-Flow problem, c.f., [10]) while both directed and undirected minimum-power variants are unlikely to have even a polylogarithmic approximation [16, 22]. Another example is finding an arborescence rooted at s, that is, a subgraph that contains an sv-path for every node v. The minimum-cost case is in P (c.f., [10]), while the minimum-power variant is at least as hard as the Set-Cover problem. For more examples see [1, 4, 22, 24].

For min-cost problems, a standard reduction from the undirected variant to the directed one is replacing every undirected edge $e = uv$ by two opposite directed edges uv, vu of the same cost as e, finding a solution D to the directed variant and take the underlying graph G of D. However, this reduction does not work for *min-power* problems. The power of G can be much larger than that of D, e.g., if D is a star. In the power model, directed and undirected variants behave rather differently, as illustrated by the following example.

Example: Suppose that we are given an instance of MPEMC and a budget P and our goal is to solve the "budgeted coverage" version of MPEMC: to cover the maximum possible demand using power at most P. We will show that this problem is harder than the Densest k-Subgraph problem, which is defined as follows: given a graph $\mathcal{G} = (V, \mathcal{E})$ and an integer k, find a subgraph of $\mathcal{G}$ with k nodes that has the maximum number of edges. The best known approximation ratio for Densest k-Subgraph is roughly $n^{-1/3}$ [13], and in spite of numerous attempts to improve it, this ratio holds for over 11 years. We prove:

Proposition 1.6 *If there exists a ρ-approximation algorithm for the budgeted coverage version of* MPEMC *with unit costs, then there exists a ρ-approximation algorithm for* Densest k-Subgraph.

Proof: Given an instance $\mathcal{G} = (V, \mathcal{E}), k$ of Densest k-Subgraph, define an instance $(\mathcal{G}, r, P)$ of budgeted coverage version of MPEMC with unit costs as follows: $r(v) = k - 1$ for all $v \in V$ and $P = k$. 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. $\square$

The most natural heuristic for approximating MPEMC is as follows. Guess opt (more precisely, using binary search, guess an almost tight lower bound on opt). Cover maximum amount of the demand within budget opt, and iterate. Proposition 1.6 shows that this strategy fails.

2 Approximating BMGC* (Proof of Lemma 1.3)

We will give a $(1-1/e)$-approximation algorithm for a generalization of BMGC*, when the nodes in B also have weights $\{w(b) : b \in B\}$, and the goal is to maximize

$$\mathsf{val}(S) = \sum_{b \in B} \min\{|\Gamma_{\mathcal{G}}(b) \cap S|, r(b)\} \cdot w(b) \ .$$

We may assume that in each part, the costs defined by the ordering of the Star-Property, are strictly increasing. Clearly, we may also assume that $c(a) \leq P$ for each $a \in A$. For $S \subseteq A$ and $b \in B$, let $r_S(b) = \max\{r(b) - |\Gamma_{\mathcal{G}}(b) \cap S|, 0\}$ be the *residual requirement* of b w.r.t. S (so $r(b) = r_{\emptyset}(b)$). $S \subseteq A$ is a feasible solution if $c(S) \leq P$ and S obeys the group constraints $|S \cap A^i| \leq 1$ for every $A^i \in \mathcal{A}$.

Our algorithm and its analysis resemble the proof of Khuller, Moss, and Naor [18] for the Budgeted Maximum Coverage problem; the main difference is that our algorithm is a local replacement algorithm, while the [18] algorithm only adds elements.

Let S satisfy the group constraints, and set $s^i = A^i \cap S$ (possibly $s^i = \emptyset$). Let $B_S = \{b \in B : r_S(b) > 0\}$ be the set of *deficient* nodes w.r.t. S. For $a \in A^i$ with $c(a) > c(s^i)$, the *density of* a *w.r.t.* S is:

$$\begin{aligned} \sigma_{c,w}(S,a) &= \frac{\mathsf{val}(S - s^i + a) - \mathsf{val}(S)}{c(a) - c(s^i)} \\ &= \frac{w((\Gamma(a) - \Gamma(s^i)) \cap B_S|)}{c(a) - c(s^i)} \end{aligned}$$

The algorithm "guesses" a set S_0 with 3 elements. The goal is to find 3 elements belonging to some optimal solution, that cover the largest demand among all 3 elements in this solution. Hence we go over all possible sets S_0 of size 3, run our algorithm for all choices of S_0 and return the best solution over all S_0. Our algorithm augments a given S_0 to a feasible solution.

Procedure GREEDY(S_0)

Initialization: $S \leftarrow S_0$, $r \leftarrow r_{S_0}$, and remove from A the parts corresponding to S_0.

While $A \neq \emptyset$ do:

1. Find $a \in A$ of maximum density, and let A^i be the part with $a \in A^i$.
2. If $c(S - s^i + a) \leq P$ then $S \leftarrow S - s^i + a$, where $s^i = A^i \cap S$ (possibly $s^i = \emptyset$).
3. $A \leftarrow A - a$.

End While

The algorithm for directed BMGC is as follows. Let $k > e$ be some fixed integer.

Algorithm for $\mathsf{BMGC*}$

1. For every feasible $S_0 \subseteq A$ with $|S_0| \leq 3$ do GREEDY(S_0).
2. Among the sets S returned, output one with maximum $\mathsf{val}(S)$.

We now prove that the approximation ratio is $(1 - 1/e)$.

Remark: It may seem that starting with some "best" triplet of elements going over all possible triplets can not have crucial effect on the ratio of the algorithm. Indeed, if the final solution is very large, three elements make little difference. However, they can make a big difference in case the final solution is very small. The goal of these three elements is to overcome a "knapsack type" difficulty the algorithm encounters. The fact that the elements have costs and the budget bound P creates a problem with the last element GREEDY tries to add. Thus, if we are able to add this element (say in the best case that adding this element leads to a cost of exactly P) there would be no need for "guessing" the "correct" first three elements. Indeed if for the next element a budget overflow occurs, but this element is added nevertheless, it is easy to see that the $1-1/e$ ratio holds. However, this can bring the cost to around $2P$. Thus, since the last element may create a budget overflow, sometimes it can not be taken. The selection of the "correct" three first elements compensate for the last element not being added. We remark that with the choice of $k = 1$ (guessing only one element) the ratio is *unbounded* and with $k = 2$ the ratio is $\frac{1}{(1-1/e)+1/2}$. So $k = 3$ is the minimum possible to get the optimal $1-1/e$ ratio (this ratio is optimal as our problem generalizes the $\mathsf{Maximum\text{-}Coverage}$ problem that admits no better than $1 - 1/e$ ratio, unless P=NP [12]).

Let OPT be an optimal solution. Clearly, if $|\mathsf{OPT}| \leq 3$ the algorithm returns an optimal solution. Henceforth assume $|\mathsf{OPT}| > 3$. Let $s_1, s_2, \ldots$ be an ordering of nodes of OPT by non-decreasing coverage value.

Consider the computation at Step 1 of the algorithm when $S_0 = \{s_1, s_2, \ldots, s_k\}$ was considered. Let $\mathsf{OPT}' = \mathsf{OPT} - S_0$ and $P' = P - c(S_0)$. Let ℓ be the number of nodes added by GREEDY to S_0 until first node from OPT' is considered but not added to S because its addition would violate the budget P; let $a \in A^i$ be this node. Let S_j be the set of the first j nodes added to S_0 by GREEDY, where we set $S_{\ell+1} = S_\ell - s^i + a$. Note that

$c(S_{\ell+1}) = c(S_\ell - s^i + a) > P'$, since a was not added. Let $\Delta_i \mathsf{val}(S) = \mathsf{val}(S_i) - \mathsf{val}(S_{i-1})$ and $\Delta_i c(S) = c(S_i) - c(S_{i-1})$, $i = 1, \ldots, \ell+1$. The following two statements can be derived from [18].

Lemma 2.1 *For each* $j = 1, \ldots, \ell+1$,

$$\frac{\Delta_j \mathsf{val}(S)}{\Delta_j c(S)} \geq \frac{\mathsf{val}(\mathsf{OPT}') - \mathsf{val}(S_{j-1})}{P'}.$$

Lemma 2.2 *For every* $j = 1, \ldots, \ell+1$

$$\mathsf{val}(S_j) \geq \left[1 - \prod_{i=1}^{j} \left(1 - \frac{\Delta_i c(S)}{P'}\right)\right] \cdot \mathsf{val}(\mathsf{OPT}').$$

Remark: Note that the Star-Property is essential in our proof. If the Star-Property does not hold then there are elements in the partial solution that can be replaced improving the density by *decreasing* the number of covered elements. This makes this proof incorrect.

Applying Lemma 2.2 with $j = \ell+1$, we get (see [18] for details):

$$\mathsf{val}(S_{\ell+1}) \geq \left(1 - \frac{1}{e}\right) \cdot \mathsf{val}(\mathsf{OPT}').$$

Let S be the set returned by GREEDY(S_0), and let $S' = S - S_0$. Then

$$\mathsf{val}(S') + \Delta_{\ell+1}\mathsf{val}(S) \geq \mathsf{val}(S_{\ell+1}) \geq \left(1 - \frac{1}{e}\right) \cdot \mathsf{val}(\mathsf{OPT}').$$

In addition, $\Delta_{\ell+1}\mathsf{val}(S) \leq \frac{1}{k}\mathsf{val}(S_0)$ by the way the nodes in OPT were ordered. Thus:

$$\begin{aligned}
\mathsf{val}(S) &= \mathsf{val}(S_0) + \mathsf{val}(S') \\
&\geq \mathsf{val}(S_0) + \left(1 - \frac{1}{e}\right) \cdot \mathsf{val}(\mathsf{OPT}') - \Delta_{\ell+1}\mathsf{val}(S) \\
&\geq \mathsf{val}(S_0) + \left(1 - \frac{1}{e}\right) \cdot \mathsf{val}(\mathsf{OPT}') - \frac{1}{k}\mathsf{val}(S_0) \\
&\geq \left(1 - \frac{1}{k}\right) \cdot \mathsf{val}(S_0) + \left(1 - \frac{1}{e}\right) \cdot \mathsf{val}(\mathsf{OPT}') \\
&\geq \left(1 - \frac{1}{e}\right) \cdot (\mathsf{val}(S_0) + \mathsf{val}(\mathsf{OPT}')) \\
&= \left(1 - \frac{1}{e}\right) \cdot (\mathsf{val}(S_0) + \mathsf{val}(\mathsf{OPT} - S_0)) \\
&= \left(1 - \frac{1}{e}\right) \cdot \mathsf{val}(\mathsf{OPT})
\end{aligned}$$

The last inequality follows from the fact that $k > e$.

The proof of Lemma 1.3 is complete.

3 Approximating MPEMC (Proof of Theorem 1.2)

3.1 Reduction to bipartite graphs

We will show an $O(\log n)$-approximation algorithm for (undirected) *bipartite* MPEMC where $\mathcal{G} = (A + B, \mathcal{E})$ is a bipartite graph and $r(a) = 0$ for every $a \in A$. The following statement shows that getting an $O(\log n)$-approximation algorithm for the bipartite MPEMC is sufficient.

Lemma 3.1 *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 $p_E(V) \leq p_{E'}(V') \leq \rho\mathsf{opt}' \leq 2\rho\mathsf{opt}$. □

3.2 Am $O(\log n)$-approximation for bipartite MPEMC

We 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 defined by $r_I(v) = \max\{r(v) - d_I(v), 0\}$. One of the main challenges is achieving the following reduction, which will be proved in the next section using our algorithm for BMGC.

Lemma 3.2 *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 for $\beta = (1 - 1/e)(1 - 1/\gamma)$ the following holds:*

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

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

Note that if $\tau < \mathsf{opt}$ the algorithm may return a edge set I that satisfies (1) and (2); if the algorithm declares "$\tau < \mathsf{opt}$" then this is correct. An $O(\log n)$-approximation algorithm for the bipartite MPEMC easily follows from Lemma 3.2:

While $r(B) > 0$ do
- Find the least integer τ so that the algorithm in Lemma 3.2 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 $(1+\gamma)\mathsf{opt}$ is added. Thus the algorithm can be implemented to run in polynomial time, and has approximation ratio $O(\log r(B)) = O(\log(n^2)) = O(\log n)$.

3.3 Proof of Lemma 3.2

Let τ be an integer and let $R = r(B) = \sum_{b \in B} r(b)$. An edge $ab \in \mathcal{E}$, $b \in B$, 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 3.3 *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. $\square$

Lemma 3.4 $p_{\mathcal{I}}(B) \leq \gamma\tau$.

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

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

$\square$

Lemmas 3.3 and 3.4 imply that we may ignore the dangerous edges and still be able to cover a constant fraction of the total demand. Once dangerous edges are ignored, the algorithm does not need to take the power incurred in B into account, as the total power of B w.r.t. all the non-dangerous edges is $\gamma\tau = O(\mathsf{opt})$. Therefore, the problem we want to solve is similar to the bipartite MPEMC, except that we want to minimize the power of A only. Formally:

Instance: A bipartite graph $\mathcal{G} = (A + B, \mathcal{I})$ with edge-costs $\{c(e) : e \in \mathcal{I}\}$, requirements $\{r(b) : b \in B\}$, and a budget $\tau = P$.
Objective: Find $I \subseteq \mathcal{I}$ with $p_I(A) \leq P$ and maximum $\sum_{b \in B} \min\{d_I(b), r(b)\}$.

Lemma 3.5 *The above problem admits a* $(1 - 1/e)$*-approximation algorithm.*

Proof: We show that the problem above can be reduced, while preserving approximation ratio, to BMGC^*. Given an instance of the above problem, construct an instance of BMGC^* as follows. For every $a \in A$ do the following. Let $e_1, ..., e_k$ be the edges incident to a sorted by increasing costs. For every e_i add a node a_i of cost $c(a_i) = c(e_i)$ and for every edge ab of cost $\leq c(e_i)$ add an edge $a_i b$. The group corresponding to $a \in A$ is $A^a = \{a_1, \ldots, a_k\}$, so $\mathcal{A} = \{A^a : a \in A\}$. Clearly, the groups are disjoint, hence we obtain a BMGC instance. The Star-Property holds by the construction. Every node in A^a corresponds an edge incident to a and has the cost of this edge; thus choosing one node from $\mathcal{A}^a$ also determines the power level of a. Thus, keeping costs, to every solutions to the obtained BMGC instance, corresponds a unique solution to the problem defined above, and vice versa. The statement now follows from Lemma 1.3. $\square$

The algorithm for Lemma 3.2 is as follows:

1. With budget τ, compute $I \subseteq \mathcal{I}$ using the $(1 - 1/e)$-approximation algorithm from Lemma 3.5.
2. *If* $r_I(B) \leq (1 - \beta)R$ (recall that $\beta = 1/2(1 - 1/\gamma)$) 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 3.3, 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 3.4. As we used budget τ, $p_I(A) \leq \tau$. Thus $p_I(V) = p_I(A) + p_I(B) \leq (1 + \gamma)\tau$.

4 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 previous heuristic, namely the Cone-Based Topology Control Heuristic CBTC of Wattenhofer et. al [26] and Li et. al [23] and Bahramgiri et. al [3]. 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 [3] 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 is with exponent $c = 2$. This 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 [26] and Cartigny et. al [6]. 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.

		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 $c = 2$.

References

[1] E. Althaus, G. Calinescu, I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment for symmetric connectivity in static ad-hoc wireless networks. *Wireless Networks*, 12(3):287–299, 2006.

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

[3] M. Bahramgiri, M. T. Hajiaghayi, and V. S. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. *Wireless Networks*, 12(2):179–188, 2006.

[4] 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.

[5] G. Calinescu and P. J. Wan. Range assignment for biconnectivity and k-edge connectivity in wireless ad hoc networks. *Mobile Networks and Applications*, 11(2):121128, 2006.

[6] J. Cartigny, D. Simplot, and I. Stojmenovic. Localized minimum-energy broadcasting in ad-hoc networks. In *INFOCOM*, 2003.

[7] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. In *APPROX*, pages 72–83, 2004.

[8] J. Cheriyan, S. Vempala, and A. Vetta. An approximation algorithm for the minimum-cost k-vertex connected subgraph. *SIAM J. 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 *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 3-approximation algorithm for finding optimum 4,5-vertex-connected spanning subgraphs. *J. of Algorithms*, 32(1):31–40, 1999.

[12] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[13] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, pages 410–421, 2001.

[14] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions–II. *Math. Prog. Study*, 8:7387, 1978.

[15] M. T. Hajiaghayi, N. Immorlica, and V. S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In *MOBICOM*, pages 300–312, 2003.

[16] M. T. Hajiaghayi, G. Kortsarz, V. S. Mirrokni, and Z. Nutov. Power optimization for connectivity problems. *Math. Program.*, 110(1):195–208, 2007. Preliminary version in IPCO 2005.

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

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

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

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

[21] G. Kortsarz and Z. Nutov. *Approximating minimum-cost connectivity problems,* Ch. 58 in Approximation algorithms and Metaheuristics, Editor T. F. Gonzalez. Chapman & Hall/CRC, 2007.

[22] Y. Lando and Z. Nutov. On minimum power connectivity problems. Manuscript, submitted for journal publication. Preliminary version in ESA 2007, pages 87-98.

[23] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. A cone-based distributed topology-control algorithm for wireless multi-hop networks. *IEEE/ACM Trans. Netw.*, 13(1):147–159, 2005.

[24] Z. Nutov. Approximating minimum power covers of intersecting families and directed connectivity problems. In *APPROX*, pages 236–247, 2006.

[25] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, 2004.

[26] R. Wattenhofer, E. L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for wireless multihop ad-hoc networks. In *INFOCOM*, pages 1388–1397, 2001.