Bounded-Hops Power Assignment in Ad-Hoc Wireless Networks

G. Calinescu S. Kapoor M. Sarwat *

Abstract

Used for topology control in ad-hoc wireless networks, Power Assignment is a family of problems, each defined by a certain connectivity constraint (such as strong connectivity) The input consists of a directed complete weighted digraph $G = (V, c)$. The *power* of a vertex u in a directed spanning subgraph H is given by $p_H(u) = \max_{uv \in E(H)} c(uv)$, and corresponds to the energy consumption required for node u to transmit to all nodes v with $uv \in E(H)$. The *power* of H is given by $p(H) = \sum_{u \in V} p_H(u)$. Power Assignment seeks to minimize $p(H)$ while H satisfies the given connectivity constraint.

Min-Power Bounded-Hops Broadcast has also as part of the input a positive integer d and an $r \in V$. The output H must be a r-rooted outgoing arborescence of depth at most d. We give an $(O(\log n), O(\log n))$ bicriteria approximation algorithm for Min-Power Bounded-Hops Broadcast: that is, our output has depth at most $O(d \log n)$ and power at most $O(\log n)$ times the optimum solution.

For the Euclidean case, when $c(u,v) = c(v,u) = ||u,v||^\kappa$ (here $||u,v||$ is the Euclidean distance and κ is a constant between 2 and 5), the output of our algorithm can be modified to give a $O((\log n)^\kappa)$ approximation ratio. Previous results for Min-Power Bounded-Hops Broadcast are only exact algorithms based on dynamic programming for case when the nodes lie on the line and $c(u,v) = c(v,u) = ||u,v||^\kappa$.

Our bicriteria results extend to Min-Power Bounded-Hops Strong Connectivity (H must have a path of at most d edges in between any two nodes) and Min-Power Bounded-Hops Symmetric Connectivity (the undirected graph having an edge uv iff H has both uv and vu is required to have diameter at most d). Previous work for Min-Power Bounded-Hops Strong Connectivity consists only of constant or better approximation for special cases of the Euclidean case.

1 Introduction

Wireless networking is increasingly being used in recent years. As the battery technology has not grown at the same pace as the electronics, energy is rapidly becoming an important issue. For the purpose of energy conservation, each node can (possibly dynamically) adjust its transmitting power, based on the distance to the receiving node and the background noise.

Due to limited range, communication is achieved through intermediate nodes relaying packets. In several applications the quality of service also becomes an issue. For example, large delays may not be acceptable, and the number of total hops taken by a packet must be bounded. This motivates the study of the problems described below.

In the most general model, a weighted directed graph $H = (V, E)$ with power requirements $c : E \to R^+$ is given by the positioning of the n wireless nodes, where $c(u,v)$ represents the power requirement for the node u to establish a unidirectional link to node v. But, reflecting the broadcast nature of ad hoc wireless networks, once a node u transmits with power $p(u)$, all nodes v with $c(u,v) \leq p(u)$ receive the signal. A function $p : V \to R^+$ is called a power assignment, and it induces a directed graph, always

*Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616. E-mail: {calinesc,kapoor,sarwmoh}@iit.edu.

denoted by $G = (V, F)$, with links uv whenever $p(u) \geq c(u,v)$. The goal of the Power Assignment problem is to minimize the total power $\sum_{v \in V} p(v)$ such that the induced digraph satisfies certain connectivity constraint. In this paper we consider three connectivity constraints:

1. Min-Power Bounded-Hops Broadcast, where the induced digraph must be an r-rooted outgoing arborescence of depth at most d. The vertex r and integer d are given as a part of the input.

2. Min-Power Bounded-Hops Strong Connectivity, where the induced digraph must have a path of at most d edges in between any two nodes, where d is given as a part of the input.

3. Min-Power Bounded-Hops Symmetric Connectivity, where the undirected graph having an edge uv if and only if the induced digraph has both uv and vu must have diameter at most d, where d is given as a part of the input.

For simplicity of exposition, we use mostly the following equivalent definition of the Power Assignment problem: given a directed spanning subgraph H, define the *power* of a vertex u as $p_H(u) = \max_{uv \in E(H)} c(uv)$ and the *power* of H as $p(H) = \sum_{u \in V} p_H(u)$. To see the equivalence, note that an optimal power assignment inducing directed spanning subgraph H never has $p(v) > \max_{uv \in E(H)} c(uv)$. Then the Power Assignment problem becomes finding the directed spanning subgraph H satisfying the connectivity constraint with minimum $p(H)$.

An important special case (which we call the *Euclidean* case) is when the input graph $G = (V, c)$ has power requirements given by $c(u,v) = c(v,u) = ||u,v||^{\kappa}$, where $||u,v||$ is the Euclidean distance and κ is a constant between 2 and 5. This case is motivated by signal transmission in a network embedded in a two-dimensional space without any obstacles [15], with κ being the path-loss exponent.

Minimizing the power contradicts bounding the number of hops in the induced subgraph, as it has been noted by [6, 12] and formal tradeoff results in a similar but different model have been obtained by [1]. Indeed, if we look at the following example in the Euclidean case: n points on the line with distance 1 in between two consecutive vertices, we note that without a bound on the number of hops Min-Power Broadcast has an optimum of $n-1$, while with a bound of one hop, the unique solution requires $(n-1)^{\kappa}$ power.

Numerous papers on Power Assignment have been published recently and we refer to [5] for a slightly out-of-date survey. We mention here only work on bounded-hops Power Assignment. Kirousis et. al. [12] consider Min-Power Bounded-Hops Strong Connectivity when then power requirements are Euclidean and the nodes are equidistant on the line. In [7] Clementi et. al. present a 2-approximation algorithm for Min-Power Bounded-Hops Strong Connectivity for the more general case when the nodes are on the line, but not equidistant. They also present better approximation bounds for the so-called *well spread* instances, when nodes' positions satisfy that the ratio of the largest inter-node distance to the smallest inter-node distance is at most a constant times the square root of the number of nodes. In the Euclidean case, [6] obtain constant ratio algorithms for Min-Power Bounded-Hops Strong Connectivity for well spread instances.

The result of [7] is also implied by the exact algorithms for Min-Power Bounded-Hops Broadcast when power requirement are Euclidean and the nodes are on the line obtained by [8] and [4]. In a related work, Sanders et. al [2] discuss running time issues in calculating exactly bounded-hops min-power paths.

Our results do not make any assumptions regarding the input power requirement digraph - the algorithm can handle asymmetric power requirements, which are motivated by the possible existence of non-uniform wireless nodes and by applications in solving Network Lifetime [3].

However, our results use the bicriteria type of approximation introduced by Marathe et. al [14], which allows for a relaxation of the constraint on the number of hops. The problem considered by [14] most related to ours is Bounded Diameter Minimum Spanning Trees (BDMST from now on): find a

minimum cost spanning tree with bounded diameter, where the diameter is given by a second *length* function.

We obtain $(O(\log n), O(\log n))$ bicriteria approximation algorithms for Min-Power Bounded-Hops Broadcast, Min-Power Bounded-Hops Strong Connectivity, and Min-Power Bounded-Hops Symmetric Connectivity. That is, our output has number of hops bounded by $O(d \log n)$ and power at most $O(\log n)$ times the optimum solution with number of hops d.

We present a straightforward reduction showing that improving our results for any of the problems studied in the case of arbitrary power requirements would lead to better ratios for BDMST when the length function is 1 on all the edges. However there has been no progress on BDMST for quite a while. Only for the Euclidean case, we can show that simple postprocessing gives an $O((\log n)^{\kappa})$-approximation algorithm for Min-Power Bounded-Hops Broadcast. Weaker similar results hold for Min-Power Bounded-Hops Strong Connectivity and Min-Power Bounded-Hops Symmetric Connectivity.

Our methods are based on the bicriteria techniques of Marathe et. al. [14], on the spider techniques of Klein and Ravi (used for Node-Weighted Steiner Tree [13]) as adapted for Power Assignment problems by [3], and on an extension of the Set Coverage problem [9].

The organization of the paper is as follows. Section 2 gives the main algorithm for Min-Power Bounded-Hops Broadcast, after introducing definitions and the reduction mentioned above. Sections 3 and 4 discussed the extension of our techniques to Bounded-Hops Min-Power Strong Connectivity and Bounded-Hops Min-Power Symmetric Connectivity. The postprocessing technique for the Euclidean case is presented in Section 5. We present conclusions and open problems in Section 6.

2 Min-Power Bounded-Hops Broadcast

We start with a lemma which shows that Min-Power Bounded-Hops Broadcast problem is atleast as hard to approximate as the conventional bicriteria problem of Bounded diameter min cost spanning tree (BDMST). Therefore, any improvement of the guarantees presented here would imply a better than known guarantee for BDMST problem. It may be mentioned here that any improvement in the BDMST guarantee has not been reported since last nine years.

Given the conventional bicriteria problem BDMST we can reduce it to a power problem by the following construction. Given a BDMST instance $G(V,E,c,d)$ with $c(e)$ giving the cost of edge e and d being the bound on the diameter, construct a Min-Power Bounded-Hops Broadcast instance as follows:

- Split each edge $e(u,v)$ into three edges $(u,u'),(u',v'),(v'v)$.
- Define $c'(u',v') = c'(v',u') = c(uv)$. Let the set of all these edges of nonzero cost be F.
- Define $c'(u,u') = c'(u',u) = c'(v',v) = c'(v,v') = 0$.
- Let $u',v' \in U'$, and $u,v \in U$
- Let $d' = 3d+1$ and the root be some vertex of V.

Claim 1: Given a tree T rooted at r in graph in G such that all nodes are within d hops from r, there is a tree T' rooted at r' in G' such that $power(T') = c(T)$ and all nodes are within $d' = 3d+1$ hops of r'.

Proof. We first show how to construct T' from a given T. For every edge in (u,v) in T, let (u,u'),(u',v') and $(v'v)$ be in T. For every node $u \in U$, there is a path from r' to u of no more than $3d$ hops. The only nodes not connected to r' in T' are in U'. But since all these nodes are connected to a node in U, they can be reached from them in 1 hop and no additional cost. Thus we can construct a solution T' from T which is of the same cost and hop distance is at most $3d+1$. □

Claim 2: Given a solution T' in G' with a diameter d', there is a solution T in G such that $cost(T) \leq cost(T)$ and diameter of T is at most $\lfloor \frac{d}{3} \rfloor$

Proof. Modify the solution as follows: For every non zero cost edge (u', v') in T include the two non zero cost edges u, u', v', v. Since we are only adding edges the hop distance from r' to any node can not increase. Now construct T as follows. For every triad of edges $(u, u'), (u', v'), (v'v)$, add the edge (u, v) to T. The cost is the same.

Let a node u in T' be d' hops away from r. The node either is not there in T, or the path has been compressed to $\lfloor \frac{d'}{3} \rfloor$. Thus all the nodes in T can be reached within $\lfloor \frac{d'}{3} \rfloor$ hops.

□

Lemma 1 *An algorithm with an approximation ratio of $(O(f(n)), O(f(n)))$, for min power broadcast implies a ratio of $(O(f(n)), O(f(n)))$ for BDMST.*

The proof is immediate from the above two claims. Given an instance $I(d, n)$ of BDMST with size n and diameter d, map it to an instance $I'(3d+1, 3n)$ as illustrated in claim 1, solve it with $(O(f(3n)), O(f(3n)))$ ratio, and convert it back to a solution to I of the same cost and and one third the diameter, implying an $(O(f(n)), O(f(n)))$ guarantee.

Now we present some definitions used for our algorithm for Min-Power Bounded-Hops Broadcast and with a straightforward reduction from BDMST to Min-Power Bounded-Hops Broadcast. By *cost* we mean power requirement $c(uv)$.

An approximation algorithm for the next problem will be used as a subroutine by our main algorithm.

Budgeted Set Coverage [11] The input consists of a collection of sets $\mathcal{R} = \{S_1, S_2, ..., S_m\}$ with associated prices $\{c_i\}_{i=1}^m$ defined over a domain of elements $X = \{x_1, x_2, ..., x_n\}$ The goal is to find a collection of sets $\mathcal{S} \subseteq \mathcal{R}$, such that the total sum of the prices of the sets in $\mathcal{S}$ does not exceed the budget B, and the total number of elements covered is maximized.

Khuller et. al. [11] presented a $1 - 1/e$-approximation algorithm for Budgeted Set Coverage. However, we are not using directly their result since in our instance has exponentially many sets.

Branch A branch is a directed graph $S = (r_S, V_S, E_S)$ such that (V_S, E_S) contains a directed path from the vertex r_S (called the root of the branch) to every vertex of V_S. A *d-bounded branch* is a branch such that every node is no more than d hops away from the root of the branch.

The main algorithm assumes an estimate B on the power of the optimum solution is given. The algorithm finds a r-rooted spanning arborescence of depth $O(d \log n)$ and power $O(B \log n)$, if a solution of power at most B exists, but might terminate with a "failure" message if no such solution exists. A simple binary search on the value of B (in case of failure, the value of B is increased, otherwise a lower value is tried) would then give a bicriteria $(O(\log n), O(\log n))$ approximation for Min-Power Bounded-Hops Broadcast.

2.1 Main Algorithm

The main algorithm is an adaptation of the approximation algorithm of Marathe et al [14] for the BDMST problem. The algorithm works in phases and maintains a set of edges Q and a set P of nodes such that all the remaining node are reachable from $P \cup \{r\}$ by paths of Q with small number of hops (the precise bound on the number of hops is given in Lemma 5).

In each phase, the algorithm the solution of a Budgeted Set Coverage instance attempting to significantly reduce $|P|$ without exceeding a certain budget.

Algorithm 1 Broadcast

Require: A directed graph $G(V,E,c)$, a diameter bound d, a budget B, and a root r.

Ensure: Outputs a spanning arborescence A rooted at r, such that all the nodes are within $O(d\log|V|)$ hops of the root r and $power(A) = O(B\log|V|)$ provided a solution of power at most B exists; may exit with "failure" if no such solution exists.

1: $P \leftarrow V \setminus \{r\}, \quad Q \leftarrow \emptyset$
2: **while** $|P| \geq 1$ **do**
3: $\quad \mathcal{S} \leftarrow$ **setCoverage**(P)
4: $\quad$ **for all** branches $H(r_H, V_H, E_H) \in \mathcal{S}$ **do**
5: $\quad\quad Q \leftarrow Q \cup E_H$
6: $\quad\quad P \leftarrow P \setminus V_H$
7: $\quad$ **end for**
8: $\quad$ **for all** $H(r_H, V_H, E_H) \in \mathcal{S} \wedge r_H \neq r$ **do**
9: $\quad\quad P \leftarrow P \cup r_H$
10: $\quad$ **end for**
11: **end while**
12: Compute and output a $O(d\log|V|)$-bounded shortest path spanning arborescence rooted at r and using only edges of Q.

Some explanation is needed for the last step. By d-bounded we mean, as usual, with a number of edges bounded by d, while shortest is with respect to the cost function c. Such shortest path arborescences can be computed by straightforward modification of Dijkstra's algorithm to compute for any $j = 0, 1, \ldots, d$ the shortest path j-bounded arborescences. This procedure is also described in [**?**, **?**, **?**]. Finally, in the last step, the constant in the big-Oh is the one given by Lemma 5.

2.2 Solving the Budgeted Set Coverage Problem

The Budgeted Set Coverage instance **setCoverage**(P) used in Step 4 of the main algorithm **Broadcast** has as elements P, a set of vertices of G. The sets are given by certain d-bounded branches, the price of a branch is its power, and a branch S_i covers a vertex v if $v \in V_{S_i}$. Precisely, we allow only *valid* branches, which are branches S with $|V_S \cap P| \geq 2$ or with $r_S = r$. Such a restriction is needed to show a significant reduction (used later in the proof of Lemma 5) of $|P|$. There are exponentially many sets in this Budgeted Set Coverage problem, and this does not allow the direct application of the algorithm from [11].

However, the approximation algorithm for Set Coverage ([10] and [9]) is greedy, and we can still use it because, assuming the optimum solution has power at most B, we can determine a set (a branch) with average power per node covered at most $B/|U|$, where U is the current set of uncovered nodes.

2.3 Determining a good valid branch

A valid branch S is *good* if and only if $p(S)/|U \cap V_S| \leq B/|U|$, a property needed later in the proof of Lemma 3. Given a graph G, a set of uncovered nodes U and a vertex v, the goal is to find a good valid branch rooted at v, if such a branch exists.

The algorithm builds branches W by first selecting a root v and a vertex w and setting W to consist of the arcs from v to those nodes u with $c(vu) \leq c(vw)$. Until a good valid branch is found, the algorithm repeatedly and greedily adds shortest paths of appropriate number of hops from covered to uncovered nodes. The algorithm uses as a subroutine **shortestBoundedPath(**y, z, i**)**, which returns the path in G from vertex y to vertex z using at most i edges and having cost c minimum. We use $d_W(v,y)$ to denote

Algorithm 2 setCoverage(P)

Require: A set of uncovered nodes $P \subseteq V \setminus \{r\}$. Implicit input parameters are the directed graph $G(V,E,c)$, diameter bound d, budget B, and root r.

Ensure: Returns a set $\mathcal{S}$ of d-bounded valid branches such that their total power is no more than $2B$ and at least $1/3$ of the nodes of P are covered, provided there is a solution to the original problem with power at most B; may exit with "failure" if no such solution exists.

$C \leftarrow 0$
$U \leftarrow P$
$\mathcal{S} \leftarrow \phi$
$i \leftarrow 0$
repeat
  $S_i \leftarrow$ **Greedy**(U)
  $U \leftarrow U \setminus V_{S_i}$
  $C \leftarrow C + p(S_i)$
  $i \leftarrow i+1$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_i\}$
until greedy returns failure **or** $C \geq B$ **or** $U = \emptyset$
if $|U| > \frac{2}{3}|P|$ **then**
  exit failure
else
  return $\mathcal{S}$
end if

the length, in number of edges, from v to y in W. The detailed algorithm is given in full below or on the next page.

2.4 Analysis and Correctness

The analysis proceeds in the reverse order, starting with the correctness of Algorithm 3 and finishing with the bounds of Algorithm 1. The next lemma is the counterpart of Lemma 4.1 and Theorem 3.1 of [13], and a variation of it is used in [3].

Lemma 2 *Assuming there is a solution to the original problem of power at most B, Algorithm 3 finds a good valid branch.*

Proof. Let T be the optimum arborescence outgoing from the root (which has depth d, and we assume T has power at most B) and U be the set of uncovered nodes. Traverse T in postorder and whenever a vertex v is the ancestor of at least two vertices of U (where by default every vertex is an ancestor of itself) define a branch with root v given by the subtree of T rooted at v. Remove v and its descendents from T, and repeat. The process stops when $|V(T) \cap U| < 2$. If $|V(T) \cap U| = 1$, one last branch is given by the root r and the current T. Note that every obtained branch is valid: it either covers two vertices of U, or it has r as a root. Moreover every branch obtained has depth at most d. Let S_i, for $1 \leq i \leq q$ be the branches so obtained. Each branch S_i is in fact a spider - the paths from the root to the vertices of outdegree zero are disjoint.

It is immediate that $p(S_1) + p(S_2) + \ldots + p(S_q) \leq B$. We have that $|U \cap V_{S_1}| + |U \cap V_{S_2}| + \ldots + |U \cap V_{S_q}| = |U|$. Therefore there is d-bounded branch (which is a spider) of optimum S with $p(S)/|U \cap V_S| \leq B/|U|$, and either $r_S = r$ or $|U \cap V_S| = 2$. Let $Q_1, Q_2, \ldots, Q_q$, where $q = |U \cap V_S|$, be the paths of S from the children of r_S (where all these paths could have zero edges, and we use the convention that

Algorithm 3 Greedy(U)

Require: Set of uncovered nodes U. Implicit input parameters are the directed graph $G(V,E,c)$, diameter bound d, budget B, and root r.

Ensure: Outputs a valid branch S such that $p(S)/|V_S \cap U| \leq B/|U|$, provided there is a solution to the original problem with power at most B; may exit with "failure" if no such solution exists.

```
for all v ∈ V do
  for all w ∈ V \ {v} do
    E_W ← {vu | u ≠ v ∧ c(vu) ≤ c(vw)}
    V_W ← {v} ∪ {u | c(vu) ≤ c(vw)}
    while [p(W)/|U ∩ V_W| ≤ B/|U| and (|U ∩ V_W| ≥ 2 or v = r)] or [U ⊆ V_W] do
      P ← (min_{y∈V_W, u∈U\V_W} shortestBoundedPath(y, u, d − d_W(v,y)))
      E_W ← E_W ∪ E_P,        V_W ← V_W ∪ V_P
    end while
    if p(W)/|U ∩ V_W| ≤ B/|U| then
      return (v, V_W, E_W)
    end if
  end for
end for
return failure
```

r_S is also a child of itself) to vertices of $U \cap V_S$. For $i = 1,2,\ldots,q$, note that the paths $Q_i's$ are edge disjoint and that $Q_i \cap U$ consists of exactly one vertex, the last point of Q_i. We denote by y_i and u_i the first vertex and last vertex of Q_i. Moreover, assume $p(Q_1) \leq p(Q_2) \leq \ldots \leq p(Q_q)$. Note that for a directed path, its cost equal its power. Let u_S be the vertex with $c(r_S w_S) = \max_{r_S v \in E(S)} c(r_S v)$. Then $p(S) = c(r_S w_S) + \sum_{i=1}^{q} p(Q_i)$,

During the execution of Algorithm 3, the case $v = r_S$ and $w = w_S$ is considered. Let $k = |U \cap (\{y \mid c(vy) \leq c(vw)\}$, where for convenience we use that $c(vv) = 0$. Let $P_1, P_2, \ldots, P_{q-k}$ be the first $q-k$ results of the $\min_{y \in V_W, u \in U \setminus V_W}$ **shortestBoundedPath**$(y,u,d-d_W(v,y))$ procedure and note that each P_i contains only one vertex from $U \setminus W$ - its last vertex - as otherwise just a part of P_i would be returned by **shortestBoundedPath**$(y,u,d-d_W(v,y))$.

For $i = 1,2,\ldots,q-k$, we have that $p(P_i) \leq p(Q_{k+i})$. Indeed, before searching for the i^{th} path, V_W does not contain all of $\{u_1,u_2,\ldots,u_{k+i}\}$ and contains all of $\{y_1,y_2,\ldots,y_q\}$. Therefore, one of $Q_1,Q_2,\ldots,Q_{k+i}$ is a candidate to be returned by $\min_{y \in V_W, u \in U \setminus V_W}$ **shortestBoundedPath**$(y,u,d-d_W(v,y))$, where we note that $d_W(v,y_i) = 1$ and each Q_i has at most $d-1$ edges. If the algorithm does not find a good valid branch earlier, after $q-k$ iterations of the **while** loop of Algorithm 3 (with $v = r_S$ and $w = w_S$), W satisfies $p(W) \leq c(vw) + \sum_{i=1}^{q-k} p(P_i) \leq c(vw) + \sum_{i=1}^{q} p(Q_i) = p(S)$ and $|V_W \cap U| \geq |V_S \cap U|$ and either $|V_W \cap U|$ or $v = r$. We conclude that the algorithm always finds a good valid branch. □

The next lemma uses the method of Hochbaum and Pathria [10] (see also [9]) designed for analyzing the greedy algorithm for Set Coverage, the simpler version of Budgeted Set Coverage where all the sets have price 1. We did not attempt to optimize the two constants in the lemma. [9].

Lemma 3 *Provided there is a solution to the original problem with power at most B, the algorithm* **setCoverage** *finds a set of valid branches of total power at most $2B$ and together covering at least $|P|/3$ nodes of P.*

Proof. Since the power per required node of a branch chosen is no more than $B/|U|$, the power of a branch can not be more than B. Hence the total power is at most $2B$, since the algorithm stops as soon as B is exceeded.

Now we show that at least $|P|/3$ nodes of P are covered. Let $S_1, S_2, \ldots, S_k$ be the branches returned by the greedy algorithm 3. Let U_1 be P and U_{i+1} be the set of uncovered nodes of P after the selection of branch S_i by the greedy algorithm. Let $q_i = |U_i \cap V_{S_i}|$. Lemma 2 implies:

$$\frac{p(S_i)}{q_i} \leq \frac{B}{|U_i} = \frac{B}{|P| - \sum_{j=1}^{i-1} q_j}.$$

Therefore:

$$q_i \geq \frac{p(S_i)}{B}\left(|P| - \sum_{j=1}^{i-1} q_j\right)$$

and after summing up these equations we obtain:

$$\sum_{i=1}^{k} q_i \geq \left(\frac{|P|}{B}\sum_{i=1}^{k} p(S_i)\right) - \sum_{i=1}^{k}\frac{p(S_i)}{B}\sum_{j=1}^{i-1} q_j = \left(\frac{|P|}{B}\sum_{i=1}^{k} p(S_i)\right) - \sum_{j=1}^{k-1}\frac{q_j}{B}\sum_{i=j+1}^{k} p(S_i)$$

Now we use $\sum_{i=1}^{k} p(S_i) \geq B$ and $\sum_{i=j+1}^{k} p(S_i) \leq 2B$ to obtain:

$$\sum_{i=1}^{k} q_i \geq |P| - \sum_{j=1}^{k-1} 2q_j$$

and therefore

$$3\sum_{i=1}^{k} q_i \geq |P|,$$

and we conclude that at least $|P|/3$ nodes of P are covered. □

We continue with the proof of the correctness of the main algorithm 1, which follows closely Marathe et. al [14].

Lemma 4 *The number of phases of Algorithm 1 is* $O(\log n)$.

Proof. In each phase, at least $1/3$ of the nodes in P are covered. Thus the total number of the nodes in the next phase is less than or equal to $2|P|/3$ uncovered nodes and at most $|P|/6$ roots of branches, as each branch whose head is not the root covers at least two nodes of P. Thus in each phase the number reduces by a fraction of $1/6$, and thus the number of phases is $O(\log n)$ as $|P|$ is n to start with. □

The next two lemmas appear basically in [14].

Lemma 5 *The depth of the solution produced by* **Broadcast** *is* $O(\log(n)d)$.

Proof. It follows immediately by induction that after phase i, for every $v \in V$, Q contains a path of length at most di from some vertex of P to v. □

Lemma 6 *The power of the output is* $O(B\log n)$.

Proof. The lemma follows immediately from the facts that there are $O(\log n)$ phases and in each we incur no more than $2B$ power. □

We now combine the above two lemmas to state the central result of this section:

Theorem 7 *Assuming there is a solution to the original problem of power at most B, the solution produced by the algorithm 1 has power* $O(B\log n)$ *and all the nodes are reachable from the root within* $O(d\log n)$ *hops*

3 Min-Power Bounded-Hops Strong Connectivity

In this section we use the result and methods of the previous section to give an approximation algorithm for the Strong Connectivity problem with asymmetric power requirements.

We again show how to solve a budgeted version of the problem. Let this optimal solution connecting all required nodes with paths of at most d hops be OPT where $p(OPT) \leq B$. Let v be an arbitrary vertex. OPT contains an outgoing arborescence of depth at most d A_{out} rooted at v (so $p(A_{out}) \leq B$) and an incoming arborescence of depth at most d A_{in} rooted at v (so $p(A_{in}) \leq OPT$).

The broadcast algorithm in the previous section produces an outgoing arborescence of depth $O(d \log n)$ Q_{out} rooted at v with $p(Q_{out}) \leq O(B \log n)$. To obtain an incoming arborescence Q_{in} of depth $O(d \log n)$ Q_{out} with $p(Q_{in}) \leq O(B \log n)$ we use exactly the same algorithm, but computing incoming branches, and a modification described below. The main algorithm 1 and the setCoverage algorithm are the same, except that incoming branches are used instead of outgoing branches. The Greedy algorithm is simpler: we only consider minimum cost path of at most d hops from representatives to the root of the branch. All the proofs follow exactly as in the previous section.

Then $p(Q_{out} \cup Q_{in}) \leq p(Q_{out}) + p(Q_{in}) \leq O(B \log n)$ and $Q_{out} \cup Q_{in}$ is a strongly connected spanning subgraph with diameter $O(d \log n)$. Therefore we have:

Theorem 8 *There is a polynomial-time algorithm for Min-Power Bounded-Hops Strong Connectivity whose output has power $O(p(OPT) \log n)$ and every node is reachable from another node by a path of length $O(d \log n)$, where OPT is an optimum solution of diameter d.*

4 Min-Power Bounded-Hops Symmetric Connectivity

A reformulation of symmetric connectivity is searching for an undirected tree T minimizing $p(T)$. Again, we use the same approach as for Min-Power Bounded-Hops Broadcast, with modifications described below.

Branches used here are rooted and bidirected, in the sense that both directed edges uv and vu must be present for every edge of the branch. In the main algorithm we pick r arbitrarily, and we construct a $O(d \log n)$-depth spanning bidirected branch rooted at r.

The greedy algorithm tries all vertices v and w as follows. It keeps and modifies a directed graph W giving a bidirected branch (initially W is just the root v) and a set of edges Q given by the arcs from v to those nodes u with $c(vu) \leq c(vw)$. We count as covered the representatives in V_W, but we use $p(E_W \cup Q)$ instead of $p(W)$ for computing the power of the returned branch.

In addition, we use a symmetric variation of **shortestBoundedPath(y, z, i)** which has four parameters: vertices x, y, z and integer i. Precisely, **symmetricBoundedPath(x,y,z,i)** finds a minimum power directed graph P which contains the edge yx and a bidirected path from y to z using at most i edges. Such a directed graph P is found by applying the method described in [**?**, **?**, **?**] for bounded-hops minimum cost directed paths to an auxiliary directed graph whose construction is given in [**?**] for Min-Power Symmetric Unicast.

symmetricBoundedPath(x,y,z,i) is called with either $x = y \in V_W$, $i = d - d_W(v,x)$ and $z \in U$ (as in Algorithm 3 of Section 2), or with $vy \in Q$, $i = d - 1$, and $z \in U$. No other modifications are done to the algorithms. All the proofs follow exactly as in Section 2, except a straightforward "symmetric" version of Lemma 2 is used. Therefore we have:

Theorem 9 *There is a polynomial-time algorithm for Min-Power Bounded-Hops Symmetric Connectivity whose output has power $O(p(OPT) \log n)$ and diameter $O(d \log n)$, where OPT is an optimum solution of diameter d.*

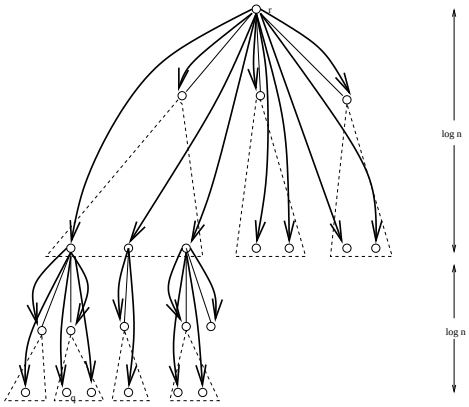

Figure 1: Adding shortcuts to reduce depth

5 Approximation for Min-Power Bounded-Hops Broadcast in the Euclidean case

The approximation factors for the broadcast problem can be improved in the case when nodes represent points on a plane, and the cost $c(uv) = c(v,u) = ||u,v||^\kappa$, where $||u,v||$ is the Euclidean distance and κ is a constant. The algorithm simply shortcuts paths of $O(\log n)$ hops (as illustrated in Figure 1) and the pseudocode of the recursive procedure is given below or on the next page. First call has $h = d$. In the algorithm, q is such that the output of the Algorithm 1 has depth qd.

Algorithm 4 Shortcut(T,h)

Require: The input is an arborescence T rooted at r with depth $h \cdot q$

Ensure: The output T' is an arborescence rooted at r, spannning $V(T)$, with depth h and $P(T') \leq p(T)q^{\kappa-1}$.

```
if h = 0 then
    return
end if
L ← {u : depth(u) = q}
M ← {u : depth(u) ≤ q}
for all u_i ∈ L do
    T_i ← subtree of T rooted at u_i
    T_i' ← Shortcut(T_i, h − 1)
end for
T' ← ∪_{u_i∈L} T_i' ∪_{u∈M\{r}} {ru}.
return T'
```

Lemma 10 *Algorithm 4 is correct, that is, T' has depth h and $p(T') \leq p(T)q^{\kappa-1}$.*

Proof. The depth property follows immediately by induction on h.

We show that $p_{T'}(r) \leq q^{\kappa-1}\sum_{u\in M\setminus L} p_T(u)$ and the rest follows by induction. Let u be such that $c(ru) = \max_{v\in M} c(rv)$ and let $P =< x_0 = r, x_1, \ldots, x_j = u >$ be the directed path in T from r to u, and note that $j \leq q$. Then $||ru||^\kappa \leq j^{\kappa-1}\sum_{i=0}^{j-1} ||x_i x_{i+1}||^\kappa$, as the worst case occurs when $||x_0x_1|| = ||x_1x_2|| = \ldots = ||x_{j-1}x_j||$.

Note that for $i = 0, 1, \ldots, j-1$ we have $p_T(x_i) \geq ||x_i x_{i+1}||^\kappa$ and that $x_i \in M \setminus L$. Therefore $p_{T'}(r) \leq q^{\kappa-1}\sum_{u\in M\setminus L} p_T(u)$ and the lemma follows. □

We know by Lemmas 5 and 6 that we can assume that the output of the algorithm of Section 2 has depth dq, with $q = O(\log n)$, and power $O(p(OPT)\log n)$, where OPT is an optimum solution of depth d. Based on the discussion above we haveL

Theorem 11 *There is a $O((\log n)^\kappa)$-approximation algorithm for Min-Power Bounded-Hops Broadcast in the Euclidean case.*

The same trick cannot be directly applied to Min-Power Bounded-Hops Strong Connectivity and Min-Power Bounded-Hops Symmetric Connectivity. However, by appropriately some special type of branches and using them in Algorithm 3 (details to appear in a full version of the paper), we claim:

Theorem 12 *In the Euclidean case, for both Min-Power Bounded-Hops Strong Connectivity and Min-Power Bounded-Hops Symmetric Connectivity. there are polynomial-time algorithms with output of diameter at most $\max(d, \log n)$ and power $O(p(OPT)(\log n)^\kappa)$, where OPT is an optimum solution of diameter d.*

6 Conclusions and Open Problems

We provided $(O(\log n), O(\log n))$ bicriteria approximation algorithms for Min-Power Bounded-Hops Broadcast, Min-Power Bounded-Hops Strong Connectivity, and Min-Power Bounded-Hops Symmetric Connectivity, with asymmetric cost function. In the Euclidean case, postprocessing the output of the bicriteria Min-Power Bounded-Hops Broadcast leads to a $O((\log n)^\kappa)$-approximation algorithm.

The Euclidean case might be easier to solve. However, it would be reasonable to start with BDMST first before moving to minimizing power. As far as we know, nothing better than $(O(\log n), O(\log n))$ is known for BDMST when hops are used for measuring diameter and simple Euclidean distance is used for measuring cost, except for the type of "jumps" we use in Section 5 to transform bicriteria results into pure approximation algorithms.

We also note that with asymmetric costs, Min-Power Multicast (even with unbounded hops) is apparently harder than Min-Power Broadcast: [3] presents a straightforward reduction from Directed Steiner Tree to Min-Power Multicast. The relevant part of [3], a paper accepted but not published, is available on the first author's web page.

The techniques seem to easily generalize to bounded degree structures, just by considering bounded degree branches (or spiders) instead of bounded depth branches in the greedy algorithm of Subsection 2.3. However, in the Euclidean case there is no need to specially search for bounded-degree trees, as the minimum spanning tree is within a constant optimum for all three Min-Power problems, and can be assumed to have degree at most five.

References

[1] F. Meyer auf der Heide, C. Schindelhauer, K. Volbert, and M Grunewald. Congestion, energy and delay in radio networks. In *SPAA*, 2002.

[2] P. Beir, P. Sanders, and N. Sivadasan. Energy optimal routing in radio networks using geometric data structrures. In *ICALP*, number 2380 in LNCS, pages 366–376, 2002.

[3] G. Calinescu, S. Kapoor, A. Olshevsky, and A. Zelikovsky. Network lifetime and power assignment in ad-hoc wireless networks. In *ESA*, 2003.

[4] I. Caragiannis, C. Kaklamanis, and P. Kanellopoulos. New results for energy-efficient broadcasting in wireless networks. In *ISAAC' 2002*, 2002.

[5] A. Clementi, G. Huiban, P. Penna, G. Rossi, and Y. Verhoeven. Some recent theoretical advances and open questions on energy consumption in ad-hoc wireless networks. In *3rd Workshop on ARACNE*, 2002.

[6] A. E. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. *Electronic Colloquium on Computational Complexity*, Report TR00-054, 2000.

[7] A.E.F. Clementi, A. Ferreira, P. Penna, S. Perennes, and R. Silvestri. The minimum range assignment problem on linear radio networks. In *ESA '00*, volume 1879 LNCS, pages 143–154, 2000.

[8] A.E.F. Clementi, M.Di Ianni, and R. Silvestri. The minimum broadcastrange assignment problem on linear multi-hop networks. In *Theoretical Comp. Sc.*, To appear.

[9] D. Hochbaum. *Approximation Algorithms for NP Hard Problems.* Brooks Cole, 1996.

[10] D. Hochbaum and A. Pathria. Analysis of the greedy approach in covering problems, 1994.

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

[12] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305, 2000.

[13] P. Klein and R.Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *Journal of Algorithms*, 19:104–115, 1995.

[14] M.V. Marathe, R. Ravi, R. Sundaram, S.S. Ravi, D.J. Rosenkrantz, and H.B. Hunt III. Bicriteri.a network design problems. In *Automata, Lang. and Prog.*, pages 487–498, 1995.

[15] Rajmohan Rajaraman. Topology control and routing in ad hoc networks: A survey. *SIGACT News*, 33:60–73, 2002.