

On Min-Power Steiner Tree*

Fabrizio Grandoni

IDSIA, University of Lugano, Switzerland, fabrizio@idsia.ch

Abstract In the classical (min-cost) Steiner tree problem, we are given an edge-weighted undirected graph and a set of terminal nodes. The goal is to compute a min-cost tree S which spans all terminals. In this paper we consider the min-power version of the problem (a.k.a. symmetric multicast), which is better suited for wireless applications. Here, the goal is to minimize the total power consumption of nodes, where the power of a node v is the maximum cost of any edge of S incident to v . Intuitively, nodes are antennas (part of which are terminals that we need to connect) and edge costs define the power to connect their endpoints via bidirectional links (so as to support protocols with ack messages). Observe that we do not require that edge costs reflect Euclidean distances between nodes: this way we can model obstacles, limited transmitting power, non-omnidirectional antennas etc. Differently from its min-cost counterpart, min-power Steiner tree is NP-hard even in the spanning tree case (a.k.a. symmetric connectivity), i.e. when all nodes are terminals. Since the power of any tree is within once and twice its cost, computing a $\rho_{st} \leq \ln(4) + \varepsilon$ [Byrka et al.'10] approximate min-cost Steiner tree provides a $2\rho_{st} < 2.78$ approximation for the problem. For min-power spanning tree the same approach provides a 2 approximation, which was improved to $5/3 + \varepsilon$ with a non-trivial approach in [Althaus et al.'06].

In this paper we present an improved approximation algorithm for min-power Steiner tree. Our result is based on two main ingredients. We present the first decomposition theorem for min-power Steiner tree, in the spirit of analogous structural results for min-cost Steiner tree and min-power spanning tree. Based on this theorem, we define a proper LP relaxation, that we exploit within the iterative randomized rounding framework in [Byrka et al.'10]. A careful analysis of the decrease of the power of nodes at each iteration provides a $3 \ln 4 - \frac{9}{4} + \varepsilon < 1.91$ approximation factor. The same approach gives an improved $1.5 + \varepsilon$ approximation for min-power spanning tree as well. This matches the approximation factor in [Nutov and Yaroshevitch'09] for the special case of min-power spanning tree with edge weights in $\{0, 1\}$.

1 Introduction

Consider the following basic problem in wireless network design. We are given a set of antennas, and we have to assign the transmitting power of each antenna. Two antennas can exchange messages (directly) if they are within the transmission range of each other (this models protocols with ack messages). The goal is to find a minimum total power assignment so that a given subset of antennas can communicate with each other (using a multi-hop protocol).

* This research is partially supported by the ERC Starting Grant NEWNET 279352.

We can formulate the above scenario as a *min-power Steiner tree* problem (a.k.a. *symmetric multicast*). Here we are given an undirected graph $G = (V, E)$, with edge costs $c : E \rightarrow \mathbb{Q}_{\geq 0}$, and a subset R of terminal nodes. The goal is to compute a Steiner tree S spanning R , of minimum *power* $p(S) := \sum_{v \in V(S)} p_S(v)$, with $p_S(v) := \max_{uv \in E(S)} \{c(uv)\}$. In words, the power of a node v with respect to tree S is the largest cost of any edge of S incident to v , and the power of S is the sum of the powers of its nodes¹. The *min-power spanning tree* problem (a.k.a. *symmetric connectivity*) is the special case of min-power Steiner tree where $R = V$, i.e. all nodes are terminals. Let us remark that, differently from part of the literature on related topics, we do not require that edge costs reflect Euclidean distances between nodes. This way, we are able to model obstacles, limited transmitting power, antennas which are not omnidirectional, etc.

The following simple approximation-preserving reduction shows that min-power Steiner tree is at least as hard to approximate as its min-cost counterpart: given a min-cost Steiner tree instance, replace each edge e with a path of 3 edges, where the boundary edges have cost zero and the middle one has cost $c(e)/2$. Hence the best we can hope for in polynomial time is a c approximation for some constant $c > 1$. It is known [1,11] that, for any tree S of cost $c(S) := \sum_{e \in E(S)} c(e)$,

$$c(S) \leq p(S) \leq 2c(S). \quad (1)$$

As a consequence, a ρ_{st} approximation for min-cost Steiner tree implies a $2\rho_{st}$ approximation for min-power Steiner tree. In particular, the recent $\ln(4) + \varepsilon < 1.39$ approximation² in [3] for the first problem, implies a 2.78 approximation for the second one: no better approximation algorithm is known to the best of our knowledge.

Differently from its min-cost version, min-power spanning tree is NP-hard (even in quite restricted subcases) [1,11]. By the above argument, a min-cost spanning tree is a 2 approximation. However, in this case non-trivial algorithms are known. A $1 + \ln 2 + \varepsilon < 1.69$ approximation is given in [5]. This was improved to $\frac{5}{3} + \varepsilon$ in [1]. If edge costs are either 0 or 1, the approximation factor can be further improved to $\frac{3}{2} + \varepsilon$ [13]. Indeed, the same factor can be achieved if edge costs are either a or b , with $0 \leq a < b$: this models nodes with two power states, low and high. All these results exploit the notion of k -decomposition. The first result is obtained with a greedy algorithm, while the latter two use (as a black box) the FPTAS in [14] for the *min-cost connected spanning hypergraph* problem in 3-hypergraphs. We will also use k -decompositions, but our algorithms are rather different (in particular, they are LP-based).

Our Results. In this paper we present an improved approximation algorithm for min-power Steiner tree.

Theorem 1. *There is an expected $3 \ln 4 - \frac{9}{4} + \varepsilon < 1.909$ approximation algorithm for min-power Steiner tree.*

¹ When S is clear from the context, we will simply write $p(v)$.

² Throughout this paper ε denotes a small positive constant.

Our result is based on two main ingredients. Informally, a k -decomposition of a Steiner tree S is a collection of (possibly overlapping) subtrees of S , each one containing at most k terminals, which together span S . The power/cost of a decomposition is the sum of the powers/costs of its components³. It is a well-known fact (see [2] and references therein) that, for any constant $\varepsilon > 0$, there is a constant $k = k(\varepsilon)$ such that there exists a k -decomposition of cost at most $(1 + \varepsilon)$ times the cost of S . A similar result holds for min-power spanning tree [1]. The first ingredient in our approximation algorithm is a similar decomposition theorem for min-power Steiner tree, which might be of independent interest. This extends the qualitative results in [1,2] since min-power Steiner tree generalizes the other two problems. However, the dependence between ε and k is worse in our construction.

Theorem 2. (Decomposition) *For any $h \geq 3$ and any Steiner tree S , there exists a h^h -decomposition of S of power at most $(1 + \frac{14}{h})p(S)$.*

Based on this theorem, we are able to compute a $1 + \varepsilon$ approximate solution for a proper component-based LP-relaxation for the problem. We exploit this relaxation within the iterative randomized rounding algorithmic framework in [3]: we sample one component with probability proportional to its fractional value, set the corresponding edge costs to zero and iterate until there exists a Steiner tree of power zero. The solution is given by the sampled components plus a subset of edges of cost zero in the original graph. A careful analysis of the decrease of node powers at each iteration provides a $3 \ln 4 - \frac{9}{4} + \varepsilon < 1.91$ approximation. We remark that, to the best of our knowledge, this is the only other known application of iterative randomized rounding to a natural problem.

The same basic approach also provides an improved approximation for min-power spanning tree.

Theorem 3. *There is an expected $\frac{3}{2} + \varepsilon$ approximation algorithm for min-power spanning tree.*

This improves on [14], and matches the approximation factor achieved in [13] (with a drastically different approach!) for the special case of 0-1 edge costs.

Preliminaries and Related Work. Min-power problems are well studied in the literature on wireless applications. Very often here one makes the assumption that nodes are points in \mathbb{R}^2 or \mathbb{R}^3 , and that edge costs reflect the Euclidean distance d between pairs of nodes, possibly according to some power law (i.e., the cost of the edge is d^c for some constant c typically between 2 and 4). This assumption is often not realistic for several reasons. First of all, due to obstacles, connecting in a direct way geographically closer nodes might be more expensive (or impossible). Second, the power of a given antenna might be upper bounded (or even lower bounded) for technological reasons. Third, antennas might not be omnidirectional. All these scenarios are captured by the undirected graph

³ Due to edge duplication, the cost of the decomposition can be larger than $c(S)$. Its power can be larger than $p(S)$ even for edge disjoint components.

model that we consider in this paper. A relevant special case of the undirected graph model is obtained by assuming that there are only two edge costs a and b , $0 \leq a < b$. This captures the practically relevant case that each node has only two power states, low and high. A typical goal is to satisfy a given connectivity requirement at minimum total power, as we assume in this paper.

Several results are known in the *asymmetric* case, where a unidirectional link is established from u to v iff v is within the transmission range of u (and possibly the vice versa does not hold). For example in the *asymmetric unicast* problem one wants to compute a min-power (or, equivalently, min-cost) directed path from node s to node t . In the *asymmetric connectivity* problem one wants to compute a min-power spanning arborescence rooted at a given root node r . This problem is NP-hard even in the 2-dimensional Euclidean case [7], and a minimum spanning tree provides a 12 approximation for the Euclidean case (while the general case is log-hard to approximate) [15]. The *asymmetric multicast* problem is the generalization of asymmetric connectivity where one wants a min-power arborescence rooted at r which contains a given set R of terminals. As observed in [1], the same approach as in [15] provides a $12\rho_{st}$ approximation for the Euclidean case, where ρ_{st} is the best-known approximation for Steiner tree in graphs. In the *complete range assignment* problem one wants to establish a strongly connected spanning subgraph. The authors of [11] present a 2-approximation which works for the undirected graph model, and show that the problem is NP-hard in the 3-dimensional Euclidean case. The NP-hardness proof was extended to 2 dimensions in [8]. Recently, the approximation factor was improved to $2 - \delta$ for a small constant $\delta > 0$ with a highly non-trivial approach [4]. The same paper presents a 1.61 approximation for edge costs in $\{a, b\}$, improving on the $9/5$ factor in [6]. For more general edge activation models and higher connectivity requirements, see e.g. [9,10,12] and references therein.

In this paper we consider the *symmetric* case, where links must be bidirectional (i.e. u and v are not adjacent if one of the two is not able to reach the other). This is used to model protocols with ack messages, where nodes which receive a message have to send back a notification. The symmetric unicast problem can be solved by applying Dijkstra's algorithm to a proper auxiliary graph [1]. The symmetric connectivity and multicast problems are equivalent to min-power spanning tree and min-power Steiner tree, respectively.

Proofs which are omitted due to lack of space will be given in the full version of the paper. The min-power Steiner tree is denoted by S^* .

2 A Decomposition Theorem for Min-Power Steiner Tree

A *k-component* is a tree which contains at most k terminals. If internal nodes are non-terminals, the component is *full*. A *k-decomposition* of a Steiner tree S over terminals R is a collection of k -components on the edges of S which span S and such that the following auxiliary *component graph* is a tree: replace each component C with a star, where the leaves are the terminals of C and the central node is a distinct, dummy non-terminal v_C . Observe that, even if the

component graph is a tree, the actual components might share edges. When the value of k is irrelevant, we simply use the terms component and decomposition. We will consider k -decompositions with $k = O(1)$. This is useful since a min-power component C on a constant number of terminals can be computed in polynomial time⁴. The assumption on the component graph is more technical, and it will be clearer later. Intuitively, when we compute a min-power component on a subset of terminals, we do not have full control on the internal structure of the component. For this reason, the connectivity requirements must be satisfied independently from that structure.

Assume w.l.o.g. that S consists of one full component. This can be enforced by appending to each terminal v a dummy node v' with a dummy edge of cost 0, and replacing v with v' in the set of terminals. Any decomposition into (full) k -components of the resulting tree can be turned into a k -decomposition of the same power for S by contracting dummy edges, and vice versa.

Next lemma shows that one can assume that the maximum degree of the components in a decomposition can be upper bounded by a constant while losing a small factor in the approximation.

Lemma 1. *For any $\Delta \geq 3$, there exists a decomposition of S of power at most $(1 + \frac{2}{\lceil \Delta/2 \rceil - 1})p(S)$ whose components have degree at most Δ .*

Proof. The rough idea is to split S at some node v not satisfying the degree constraint, so that the duplicated copies of v in each obtained component have degree (less than) Δ . Then we add a few paths between components, so that the component graph remains a tree. All the components but one will satisfy the degree constraint: we iterate the process on the latter component.

In more detail, choose any leaf node r as a *root*. The decomposition initially consists of S only. We maintain the invariant that all the components but possibly the component C_r containing r have degree at most Δ . Assume that C_r has degree larger than Δ (otherwise, we are done). Consider any *split node* v of degree $d(v) = d + 1 \geq \Delta + 1$ such that all its descendants have degree at most Δ . Let u_1, \dots, u_d be the children of v , in increasing order of $c(vu_i)$. Define $\Delta' := \lceil \Delta/2 \rceil \in [2, \Delta - 1]$. We partition the u_i 's by iteratively removing the first Δ' children, until there are at most $\Delta - 2$ children left: let V_1, \dots, V_h be the resulting subsets of children. In particular, for $i < h$, $V_i = \{u_{(i-1)\Delta'+1}, \dots, u_{i\Delta'}\}$. For $i = 1, \dots, h - 1$, we let C_i be a new component induced by v , V_i , and the descendants of V_i . The new root component C_h is obtained by removing from C_r the nodes $\cup_{i < h} V(C_i) - \{v\}$ and the corresponding edges. In order to maintain the connectivity of the component graph (which might be lost at this point), we expand C_{i+1} , $i \geq 1$, as follows: let P_j be any path from v to some leaf

⁴ One can guess (by exhaustive enumeration) the non-terminal nodes of degree at least 3 in C , and the structure of the tree where non-terminals of degree 2 are contracted. Each edge vu of the contracted tree corresponds to a path P whose internal nodes are non-terminals of degree 2: after guessing the boundary edges uu' and $v'v$ of P (which might affect the power of u and v , respectively), the rest of P is w.l.o.g. a min-power path between u' and v' (which can be computed in polynomial time [1]).

which starts with edge vu_j . We append to C_{i+1} the path $P_{m(i)}$ which minimizes $p(P_j) - c(vu_j)$ over $j \in V_i$. After this step, the component graph is a tree. The invariant is maintained: in fact, the degree of any node other than v can only decrease. In each C_i , $i < h$, v has degree either Δ' or $\Delta' + 1$, which is within 2 and Δ . Since $\Delta - 2 - \Delta' < |V_h| \leq \Delta - 2$, the cardinality $|V_h| + 2$ of v in C_h is also in $[2, \Delta]$. By the choice of v , all the components but C_r have maximum degree Δ . Observe that C_r loses at least $\Delta' - 1 \geq 1$ nodes, hence the process halts.

In order to bound the power of the final decomposition, we use the following charging argument, consisting of two charging rules. When we split C_r at a given node v , the power of v remains $p(v)$ in C_h and becomes $c(vu_{i\Delta'})$ in the other C_i 's. We evenly charge the extra power $c(vu_{i\Delta'})$ to nodes V_{i+1} : observe that each $u_j \in V_{i+1}$ is charged by $\frac{c(vu_{i\Delta'})}{|V_{i+1}|} \leq \frac{c(vu_{i\Delta'})}{\Delta'-1} \leq \frac{c(vu_j)}{\Delta'-1} \leq \frac{p(u_j)}{\Delta'-1}$.

Furthermore, we have an extra increase of the power by $p(P_{m(i)}) - c(vu_{m(i)})$ for every $i < h$: this is charged to the nodes of the paths $P_j - \{v\}$ with $u_j \in V_i - \{u_{m(i)}\}$, in such a way that no node w is charged by more than $\frac{1}{\Delta'-1}p(w)$. This is possible since there are $\Delta' - 1$ such paths, and the nodes of each such path have total power at least $p(P_{m(i)}) - c(vu_{m(i)})$ by construction.

Each node w can be charged with the second charging rule at most once, since when this happens w is removed from C_r and not considered any longer. When w is charged with the first charging rule, it must be a child of some split node v . Since no node is a split node more than once, also in this case we charge w at most once. Altogether, each node v is charged by at most $\frac{2}{\Delta'-1}p(v)$. \square

Proof. (Theorem 2) Apply Lemma 1 with $\Delta = h$ to S , hence obtaining a decomposition of power at most $\frac{\lceil h/2 \rceil + 1}{\lceil h/2 \rceil - 1}p(S)$ whose components have degree at most h . We describe an h^h decomposition of each such component C with more than h^h terminals. Root C at any non-terminal r , and contract internal nodes (other than r) of degree 2. For any internal node v of C , let $P(v)$ be the path from v to its rightmost child r_v , and then from r_v to some leaf terminal $\ell(v)$ using the leftmost possible path. Observe that paths $P(v)$ are edge disjoint. Pick a value $q \in \{0, 1, \dots, h-1\}$ uniformly at random, and mark the nodes at level $\ell = q \pmod{h}$. Consider the partition of C into edge-disjoint subtrees T which is induced by the marked levels⁵. Finally, for each such subtree T , we append to each leaf v of T the path $P(v)$: this defines a component C_T .

Trees T have at most h^h leaves: hence components C_T contain at most h^h terminals each. Observe that the component graph remains a tree. In order to bound the power of components C_T , note that each node u in the original tree has in each component a power not larger than the original power $p(u)$: hence it is sufficient to bound the expected number μ_u of components a node u belongs to. Suppose u is contracted or a leaf node. Then u is contained in precisely the same components as some edge e . This edge belongs deterministically to one subtree T (hence to C_T), and possibly to another component $C_{T'}$ if the node v with $e \in P(v)$ is marked: the latter event happens with probability $1/h$. Hence in this case $\mu_u \leq 1 + 1/h$. For each other node u , observe that u belongs to

⁵ Subtrees consisting of one (leaf) node can be neglected.

one subtree T if it is not marked, and to at most two such subtrees otherwise. Furthermore, it might belong to one extra component $C_{T'}$ if the node v with $ul_u \in P(v)$ is marked, where l_u is the leftmost child of u . Hence, $\mu_u \leq 1 + 2/h$ in this case. Altogether, the decomposition of C has power at most $(1 + 2/h)p(C)$ in expectation.

From the above discussion, there exists (deterministically) an h^h decomposition of power at most $\frac{\lceil h/2 \rceil + 1}{\lceil h/2 \rceil - 1} (1 + \frac{2}{h})p(S) \leq (1 + \frac{14}{h})p(S)$. \square

We remark that for both min-cost Steiner tree and min-power spanning tree (which are special cases of min-power Steiner tree), improved $(1 + \frac{O(1)}{h})$ approximate c^h decompositions, $c = O(1)$, are known [1,2]. Finding a similar result for min-power Steiner tree, if possible, is an interesting open problem in our opinion (even if it would not directly imply any improvement of our approximation factor).

3 An Iterative Randomized Rounding Algorithm

In this section we present an improved approximation algorithm for min-power Steiner tree. Our approach is highly indebted to [3]. We consider the following LP relaxation for the problem:

$$\begin{aligned}
\min \quad & \sum_{(Q,s): s \in Q \subseteq R} p_Q \cdot x_{Q,s} && (LP_{pow}) \\
\text{s.t.} \quad & \sum_{\substack{(Q,s): s \in Q \subseteq R, \\ s \notin W, Q \cap W \neq \emptyset}} x_{Q,s} \geq 1, && \forall \emptyset \neq W \subseteq R - \{r\}; \\
& x_{Q,s} \geq 0, && \forall s \in Q \subseteq R.
\end{aligned}$$

Here r is an arbitrary *root* terminal. There is a variable $x_{Q,s}$ for each subset of terminals Q and for each $s \in Q$: the associated coefficient p_Q is the power of a min-power component C_Q on terminals Q . In particular, $S^* = C_R$ induces a feasible integral solution (where the only non-zero variable is $x_{R,r} = 1$). Let $C_{Q,s}$ be the *directed component* which is obtained by directing the edges of C_Q towards s . For a fractional solution x , let us define a directed capacity reservation by considering each (Q,s) , and increasing by $x_{Q,s}$ the capacity of the edges in $C_{Q,s}$. Then the cut constraints ensure that each terminal is able to send one unit of (splittable) flow to the root without exceeding the mentioned capacity reservation. We remark that the authors of [3] consider essentially the same LP, the main difference being that p_Q is replaced by the cost c_Q of a min-cost component on terminals Q ⁶. In particular, the set of constraints in their LP is the same as in LP_{pow} . This allows us to reuse part of their results and techniques, which rely only on the properties of the set of constraints⁷.

⁶ Another technical difference w.r.t. [3] is that they consider only full components: this has no substantial impact on their analysis, and allows us to address the Steiner and spanning tree cases in a unified way.

⁷ Incidentally, this observation might be used to address also other variants of the Steiner tree problem, with different objective functions.

Algorithm 1 An iterative randomized rounding approximation algorithm for min-power Steiner tree.

- (1) For $t = 1, 2, \dots$
 - (1a) Compute a $1+\varepsilon$ approximate solution x^t to LP_{pow} (w.r.t. the current instance).
 - (1b) Sample one component C^t , where $C^t = C_Q$ with probability $\sum_{s \in Q} x_{Q,s}^t / \sum_{(Q',s')} x_{Q',s'}^t$. Set to zero the cost of the edges in C^t and update LP_{pow} .
 - (1c) If there exists a Steiner tree of power zero, return it and halt.
-

Given Theorem 2, the proof of the following lemma follows along the same line as in [3].

Lemma 2. *For any constant $\varepsilon > 0$, a $1 + \varepsilon$ approximate solution to LP_{pow} can be computed in polynomial time.*

We exploit LP_{pow} within the iterative randomized rounding framework in [3]. Our algorithm (see also Algorithm 1) consists of a set of iterations. At each iteration t we compute a $(1+\varepsilon)$ -approximate solution to LP_{pow} , and then sample one component $C^t = C_Q$ with probability proportional to $\sum_{s \in Q} x_{Q,s}^t$. We set to zero the cost of the edges of C^t in the graph, updating LP_{pow} consequently⁸. The algorithm halts when there exists a Steiner tree of cost (and power) zero: this halting condition can be checked in polynomial time.

Lemma 3. *Algorithm 1 halts in a polynomial number of rounds in expectation.*

4 The Approximation Factor

In this section we bound the approximation factor of Algorithm 1, both in the general and in the spanning tree case. Following [3], in order to simplify the analysis let us consider the following variant of the algorithm. We introduce a dummy variable $x_{r,r}$ with $p_r = 0$ corresponding to a dummy component containing the root only, and fix $x_{r,r}$ so that the sum of the x 's is some fixed value M in all the iterations. For $M = n^{O(1)}$ large enough, this has no impact on the power of the solution nor on the behaviour of the algorithm (since sampling the dummy component has no effect). Furthermore, we let the algorithm run forever (at some point it will always sample components of power zero).

Let S^t be the min-power Steiner tree at the beginning of iteration t (in particular, $S^1 = S^*$). For a given sampled component C^t , we let $p(C^t)$ be its power in the considered iteration. We define similarly $p(S^t)$ and the corresponding cost

⁸ In the original algorithm in [3], the authors contract components rather than setting to zero the cost of their edges. Our variant has no substantial impact on their analysis, but it is crucial for us since contracting one edge (even if it has cost zero) can decrease the power of the solution.

$c(S^t)$. The expected approximation factor of the algorithm is bounded by:

$$\frac{1}{p(S^*)} \sum_t E[p(C^t)] = \frac{1}{p(S^*)} \sum_t \sum_{(Q,s)} E[\frac{x_{Q,s}^t}{M} p_Q] \leq \frac{1+\varepsilon}{Mp(S^*)} \sum_t E[p(S^t)]. \quad (2)$$

Hence, it is sufficient to provide a good upper bound on $E[p(S^t)]$. We exploit the following high-level (ideal) procedure. We start from $\tilde{S} = S^*$, and at each iteration t we add the sampled component C^t to \tilde{S} and delete some *bridge* edges B^t in $E(\tilde{S}) \cap E(S^*)$ in order to remove cycles (while maintaining terminal connectivity). By construction, \tilde{S} is a feasible Steiner tree at any time. Furthermore, the power of \tilde{S} at the beginning of iteration t is equal to the power $p(U^t)$ of the forest of non-deleted edges U^t of S^* at the beginning of the same iteration⁹. In particular, $p(S^t) \leq p(U^t) = \sum_v p_{U^t}(v)$.

At this point our analysis deviates (and gets slightly more involved) w.r.t. [3]: in that paper the authors study the expected number of iterations before a given (single) edge is deleted. We rather need to study the behavior of collections of edges incident to a given node v . In more detail, let $e_v^1, \dots, e_v^{d(v)}$ be the edges of S^* incident to v , in decreasing order of cost $c_v^1 \geq c_v^2 \geq \dots \geq c_v^{d(v)}$ (breaking ties arbitrarily). Observe that $p_{U^t}(v) = c_v^i$ during the iterations when all edges e_v^1, \dots, e_v^{i-1} are deleted and e_v^i is still non-deleted. Define δ_v^i as the expected number of iterations before all edges e_v^1, \dots, e_v^i are deleted. For notational convenience, define also $\delta_v^0 = c_v^{d(v)+1} = 0$. Then

$$E[\sum_t p_{U^t}(v)] = \sum_{i=1}^{d(v)} c_v^i (\delta_v^i - \delta_v^{i-1}) = \sum_{i=1}^{d(v)} \delta_v^i (c_v^i - c_v^{i+1}). \quad (3)$$

We will provide a feasible upper bound δ^i on δ_v^i for all v (for a proper choice of the bridge edges B^t) with the following two properties for all i :

$$\text{(a)} \quad \delta^i \leq \delta^{i+1} \qquad \text{(b)} \quad \delta^i - \delta^{i-1} \geq \delta^{i+1} - \delta^i.$$

In words, the δ^i 's are increasing (which is intuitive since one considers larger sets of edges) but at decreasing speed. Consequently, from (3) one obtains

$$E[\sum_t p_{U^t}(v)] \leq \delta^1 c_v^1 + \max_{i \geq 2} \{\delta^i - \delta^{i-1}\} \sum_{i=2}^{d(v)} c_v^i = \delta^1 c_v^1 + (\delta^2 - \delta^1) \sum_{i=2}^{d(v)} c_v^i. \quad (4)$$

Inspired by (4), we introduce the following classification of the edges of S^* . We say that the power of node v is *defined* by e_v^1 . We partition the edges of S^* into the *heavy* edges H which define the power of both their endpoints, the *middle* edges M which define the power of exactly one endpoint, and the remaining *light* edges L which do not define the power of any node. Let $c(H) = \gamma_H c(S^*)$ and $c(M) = \gamma_M c(S^*)$. Observe that $p(S^*) = \alpha c(S^*)$ where $\alpha = 2\gamma_H + \gamma_M \in [1, 2]$. Note also that in (4) heavy edges appear twice with coefficient δ^1 , middle edges

⁹ Since edge weights of sampled components are set to zero, any bridge edge can be replaced by a path of zero cost edges which provides the same connectivity.

appear once with coefficient δ^1 and once with coefficient $\delta^2 - \delta^1$, and light edges appear twice with coefficient $\delta^2 - \delta^1$. Therefore one obtains

$$\begin{aligned} E[\sum_t p(U^t)] &= \sum_v E[\sum_t p^t(v)] \leq 2\delta^1 c(H) + (\delta^1 + \delta^2 - \delta^1)c(M) + 2(\delta^2 - \delta^1)c(L) \\ &= (2\delta^1 \gamma_H + \delta^2 \gamma_M + 2(\delta^2 - \delta^1)(1 - \gamma_H - \gamma_M)) \cdot c(S^*) \\ &= (2(\delta^2 - \delta^1) + (2\delta^1 - \delta^2)\alpha) \cdot \frac{p(S^*)}{\alpha} \stackrel{\alpha \geq 1}{\leq} \delta^2 p(S^*). \end{aligned} \quad (5)$$

Summarizing the above discussion, the approximation factor of the algorithm can be bounded by

$$\frac{1 + \varepsilon}{Mp(S^*)} \sum_t E[p(S^t)] \leq \frac{1 + \varepsilon}{Mp(S^*)} \sum_t E[p(U^t)] \stackrel{(5)}{\leq} \frac{(1 + \varepsilon)\delta^2}{M}. \quad (6)$$

We next provide the mentioned bounds δ^i satisfying Properties (a) and (b): we start with the spanning tree case and then move to the more complex and technical general case.

4.1 The Spanning Tree Case

Observe that in this case the optimal solution $T^* := S^*$ is by definition a terminal spanning tree (i.e. a Steiner tree without Steiner nodes). Therefore we can directly exploit the following claim in [3].

Lemma 4. [3] *Let T^* be any terminal Steiner tree. Set $\tilde{T} := T^*$ and consider the following process. For $t = 1, 2, \dots$: (a) Take any feasible solution x^t to LP_{pow} ; (b) Sample one component $C^t = C_Q$ with probability proportional to variables $x_{Q,s}^t$; (c) Delete a subset of bridge edges B^t from $E(\tilde{T}) \cap E(T^*)$ so that all the terminals remain connected in $\tilde{T} - B^t \cup C^t$. There exists a randomized procedure to choose the B^t 's so that any $W \subseteq E(T^*)$ is deleted after $M H_{|W|}$ iterations in expectation¹⁰.*

By Lemma 4 with $W = \{e_v^1, \dots, e_v^i\}$, we can choose $\delta^i = M \cdot H_i$. Observe that these δ^i 's satisfy Properties (a) and (b) since $\frac{\delta^{i+1} - \delta^i}{M} = \frac{1}{i+1}$ is a positive decreasing function of i . Theorem 3 immediately follows by (6) since $\frac{(1+\varepsilon)\delta^2}{M} = \frac{(1+\varepsilon)M H_2}{M} = (1 + \varepsilon) \cdot \frac{3}{2}$.

4.2 The Steiner Tree Case

Here we cannot directly apply Lemma 4 since S^* might not be a terminal spanning tree: w.l.o.g. assume that S^* consists of one full component. Following [3], we define a proper auxiliary terminal spanning tree T^* , the *witness tree*. We turn S^* into a rooted binary tree S_{bin}^* as follows: Split one edge, and root the tree at the newly created node r . Split internal nodes of degree larger than 3

¹⁰ $H_q := \sum_{i=1}^q \frac{1}{i}$ is the q -th harmonic number.

by introducing dummy nodes and dummy edges of cost zero. We make the extra assumption¹¹, that we perform the latter step so that the i most expensive edges incident to a given node appear in the highest possible (consecutive) levels of S_{bin}^* . Finally, shortcut internal nodes of degree 2. Tree T^* is constructed as follows. For each internal node v in S_{bin}^* with children u and z , mark uniformly at random exactly one of the two edges vu and vz . Given two terminals r' and r'' , add $r'r''$ to T^* iff the path between r' and r'' in S_{bin}^* contains exactly one marked edge. We associate to each edge $f' \in E(S_{bin}^*)$ a (non-empty) *witness set* $W(f')$ of edges of T^* as follows: $e = uv \in E(T^*)$ belongs to $W(f')$ iff the path between u and v in S_{bin}^* contains f' . There is a many-to-one correspondence from each $f \in E(S^*)$ to some $f' \in E(S_{bin}^*)$: we let $W(f) := W(f')$.

We next apply the same deletion procedure as in Lemma 4 to T^* . When all the edges in $W(f)$ are deleted, we remove f from S^* : this process defines the bridge edges B^t that we remove from \tilde{S} at any iteration. As shown in [3], the non-deleted edges U^t of S^* at the beginning of iteration t plus the components which are sampled in the previous iterations induce (deterministically) a feasible Steiner tree. Hence also in this case we can exploit the upper bound $p(S^t) \leq p(U^t) = \sum_v p_{U^t}(v)$. Let us define $W^i(v) := \cup_{j=1}^i W(e_v^j)$. In particular, in order to delete all the edges e_v^1, \dots, e_v^i we need to delete $W^i(v)$ from T^* . The next technical lemma provides a bound on δ_v^i by combining Lemma 4 with an analysis of the distribution of $|W^i(v)|$. The crucial intuition here is that sets $W(e_v^j)$ are strongly correlated and hence $|W^i(v)|$ tends to be much smaller than $\sum_{j=1}^i |W(e_v^j)|$.

Lemma 5. $\delta_v^i \leq \delta^i := \frac{1}{2^i} MH_i + (1 - \frac{1}{2^i}) \sum_{q \geq 1} \frac{1}{2^q} MH_{q+i}$.

Proof. Let us assume that v has degree $d(v) \geq 3$ and that $i < d(v)$, the other cases being analogous and simpler. Recall that we need to delete all the edges in $W^i(v)$ in order to delete e_v^1, \dots, e_v^i , and this takes time $M H_{|W^i(v)|}$ in expectation by Lemma 4. Let us study the distribution of $|W^i(v)|$. Consider the subtree T' of S_{bin}^* given by (the edges corresponding to) e_v^1, \dots, e_v^i plus their sibling (possibly dummy) edges. Observe that, by our assumption on the structure of S_{bin}^* , this tree has $i + 1$ leaves and height i . We expand T' by appending to each leaf v of T' the only path of unmarked edges from v down to some leaf $\ell(v)$ of S_{bin}^* : let C' be the resulting tree (with $i + 1$ leaves). Each edge of T^* with both endpoints in (the leaves of) C' is a witness edge in $W^i(v)$. The number of these edges is at most i since the witness tree is acyclic: assume pessimistically that they are exactly i . Let r' be the root of T' , and s' be the (only) leaf of T' such that the edges on the path from r' to s' are unmarked. Let also d' be the only leaf of T' which is not the endpoint of any e_v^j , $j \leq i$ (d' is defined since $i < d(v)$). Observe that $Pr[s' = d'] = 1/2^i$ since this event happens only if the i edges along the path from r' to d' are unmarked. When $s' \neq d'$, there are at most $|E(P')| + 1$ extra edges in $W^i(v)$, where P' is a maximal path of unmarked edges starting from r' and going to the root of S_{bin}^* . If $h_{i,v}$ is the maximum value of $|E(P')|$,

¹¹ This is irrelevant for [3], but it is useful in the proof of Lemma 5.

then $Pr[|E(P')| = q] = 1/2^{\min\{q+1, h_{i,v}\}}$ for $q \in [0, h_{i,v}]$. Altogether:

$$\delta_v^i \stackrel{\text{Lem.4}}{\leq} \sum_{g \geq 1} Pr[|W^i(v)| = g] \cdot MH_g \leq \frac{MH_i}{2^i} + \left(1 - \frac{1}{2^i}\right) \cdot \sum_{q=0}^{h_{i,v}} \frac{MH_{i+q+1}}{2^{\min\{q+1, h_{i,v}\}}} \leq \delta^i. \quad \square$$

The reader may check that the above δ^i 's satisfy Properties (a) and (b) since

$$\frac{\delta^{i+1} - \delta^i}{M} = \frac{1}{(i+1)2^{i+1}} + \left(1 - \frac{1}{2^{i+1}}\right) \sum_{q \geq 1} \frac{1}{2^q(q+i+1)} + \frac{1}{2^{i+1}} \left(\sum_{q \geq 1} \frac{H_{q+i}}{2^q} - H_i\right)$$

is a positive decreasing function of i . Theorem 1 immediately follows from (6) and Lemma 5 since $\delta^2 = \frac{H_2}{4} + 3\left(\sum_{q \geq 1} \frac{H_q}{2^q} - \frac{H_1}{2} - \frac{H_2}{4}\right) = 3 \ln 4 - \frac{9}{4}$.

Acknowledgments. We thank Marek Cygan for reading a preliminary version of this paper and Zeev Nutov for very helpful and inspiring discussions.

References

1. E. Althaus, G. Calinescu, I. I. Mandoiu, S. K. 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. A. Borchers and D.-Z. Du. The k-Steiner ratio in graphs. *SIAM Journal on Computing*, 26(3):857–869, 1997.
3. J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. An improved LP-based approximation for Steiner tree. In *STOC*, 583–592, 2010.
4. G. Calinescu. Min-power strong connectivity. In *APPROX-RANDOM*, 67–80, 2010.
5. G. Calinescu, I. I. Mandoiu, and A. Zelikovsky. Symmetric connectivity with minimum power consumption in radio networks. In *IFIP TCS*, 119–130, 2002.
6. P. Carmi and M. J. Katz. Power assignment in radio networks with two power levels. *Algorithmica*, 47(2):183–201, 2007.
7. A. E. F. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the complexity of computing minimum energy consumption broadcast subgraphs. In *STACS*, 121–131, 2001.
8. A. E. F. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. In *ECCC*, 2000.
9. M. T. Hajiaghayi, N. Immerlica, and V. S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. *Transactions on Networking*, 15(6):1345–1358, 2007.
10. M. T. Hajiaghayi, G. Kortsarz, V. S. Mirrokni, and Z. Nutov. Power optimization for connectivity problems. *Mathematical Programming*, 110(1):195–208, 2007.
11. L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305, 2000.
12. Z. Nutov. Survivable network activation problems. In *LATIN*, 594–605, 2012.
13. Z. Nutov and A. Yaroshevitch. Wireless network design via 3-decompositions. *Information Processing Letters*, 109(19):1136–1140, 2009.
14. H. J. Prömel and A. Steger. A new approximation algorithm for the Steiner tree problem with performance ratio 5/3. *Journal of Algorithms*, 36(1):89–101, 2000.
15. P.-J. Wan, G. Calinescu, X.-Y. Li, and O. Frieder. Minimum energy broadcast routing in static ad hoc wireless networks. In *INFOCOM*, 1162–1171, 2001.