Min-Power Strong Connectivity

G. Calinescu ⋆

Abstract. Given a directed simple graph $G = (V, E)$ and a cost function $c : E \to R_+$, 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 wireless 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 some connectivity constraint. In this paper, we assume E is bidirected (for every directed edge $e \in E$, the opposite edge exists and has the same cost), while H is required to be strongly connected. This is the original power assignment problem introduce in 1989 and since then the best known approximation ratio is 2 and is achieved by a bidirected minimum spanning tree. We improve this to $2 - \epsilon$ for a small $\epsilon > 0$. We do this by combining techniques from Robins-Zelikovsky (2000) for Steiner Tree, Christofides (1976) for Metric Travelling Salesman, and Caragiannis, Flammini, and Moscardelli (2007) for the broadcast version of Power Assignment, together with a novel property on T-joins in certain two-edge-connected hypergraphs. With the restriction that $c : E \to \{A, B\}$, where $0 \le A < B$, we improve the best known approximation ratio from 1.8 to $\pi^2/6 - 1/36 + \epsilon \le 1.61$ using an adaptation of the algorithm developed by Khuller, Raghavachari, and Young (1995,1996) for (unweighted) Minimum Strongly Connected Subgraph.

1 Introduction

There has been a surge of research in Power Assignment problems since 2000 (among the earlier papers are [23, 29, 14]) This class of problems take as input a directed simple graph $G = (V, E)$ and a cost function $c : E \to R_+$. The *power* of a vertex u in a directed spanning subgraph H of G is given by $p_H(u) = \max_{uv \in E(H)} c(uv)$, and corresponds to the energy consumption required for wireless node u to transmit to all nodes v with $uv \in E(H)$. The *power* (or total power) of H is given by $p(H) = \sum_{u \in V} p_H(u)$.

The study of the min-power power assignment was started by Chen and Huang [6], which consider, as we do, the case when E is bidirected (the case is sometimes called "symmetric" or "undirected" in the literature) while H is required to be strongly connected. We call this problem *Min-Power Strong Connectivity*. We use with the same name both the (bi)directed and the undirected version of G. [6] prove that the bidirected version of a minimum (cost) spanning tree (MST) of the input graph G has power at most twice the optimum, and

⋆ Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA. calinescu@iit.edu. Research supported in part by NSF grant CNS-0916743

therefore the MST algorithm has approximation ratio at most 2. This is known to be tight (see Section 2).

We improve this to $2-\epsilon$ for a small $\epsilon > 0$. We do this by combining techniques from Robins-Zelikovsky [24, 25] for Steiner Tree, Christofides [7] for Metric Travelling Salesman, Caragiannis, Flammini, and Moscardelli [4] for the broadcast version of symmetric Power Assignment (assuming a bidirected $G = (V, E, c)$ and a "root" $u \in V$ is given, H must contain a path from u to every other vertex of G), together with what we believe to be a new property on T-joins in certain two-edge-connected hypergraphs. Familiarity with the fundamental NP-Hard optimization problems Steiner Tree and Travelling Salesman will help the reader make sense of this extended abstract; however all our proofs are self-contained and only refer to previous papers for intuition.

Carmi and Katz [5] consider the case $c : E \to \{A, B\}$, where $0 \leq A < B$. It is easy to see that for minimizing total power, the hardest case is $A = 0$ and $B = 1$. Then we have to minimize the number of nodes with power B. [5] gives a 9/5 approximation and a faster 11/6 approximation algorithm. We adapt the algorithms of Khuller, Raghavachari, and Young [17, 18] for (unweighted) Minimum Strongly Connected Subgraph (given a strongly connected directed graph $G = (V, E)$, find minimum-size $F \subseteq E$ with (V, F) strongly connected) to obtain a $\pi^2/6 - 1/36 + \epsilon \leq 1.61$ approximation in this case. We also show that the algorithms of Carmi and Katz have approximation ratio exactly 7/4.

Very restricted versions of Min-Power Strong Connectivity have been proven NP-Hard [19, 8, 5]. We are not aware of better than a factor of 2 approximation except for [5], [2] (where c is assumed to be a metric), and the exact (dynamic programming) algorithms [19] for the specific case where each vertex of G maps to a point on a line, and $c(uv)$ is an increasing function of the Euclidean distance between the images of u and v, A related version, also NP-Hard, asks for H to be bidirected (also known as "undirected" or "symmetric"). This problem is called Min-Power Symmetric Connectivity, and the best known ratio of $5/3 + \epsilon$ [1] is obtained with techniques first applied to Steiner Tree; when $c : E \to \{A, B\}$ one gets 3/2 with the same method [22]. In fact, many but not all power assignment algorithms use techniques from Steiner Tree variants (or direct reduction to Steiner Tree variants; these connections to Steiner Tree are not obvious and cannot be easily explained), and in particular Caragiannis et al [4] uses the relative greedy heuristic of Zelikovsky [30]. New interesting techniques were also developed for power assignment problem, as in [20], an improvement over [15].

The existing lower bound of the optimum, which we use, is the cost of the minimum spanning tree of G. Indeed, the optimum solution OPT contains an in-arborescence rooted at v, for some $v \in V$, and then, for all $u \in V \setminus \{v\}$, $p_{OPT}(u)$ is at least the cost of the directed edge connecting u to its parent in the arborescence.

We also use a relative greedy method as in [30, 24]; Robins-Zelikovsky [24] is rarely used as a technique, and not by only citing the ratio (improved by now in [3]). We use the natural structures of [4] to improve over the minimum spanning tree. We are at a disadvantage as our new lower bound is not far from optimum (and the algorithm more than adds the two lower bounds); still the greedy Robins-Zelikovsky method allows an improvement over the factor of 2.

The new lower bound resembles the T-joins implicit in Christofides' approximation algorithm for Traveling Salesman Problem, however we have hypergraphs rather than graphs. Precisely, we are given an edge-weighted hypergraph $K = (V_K, E_K)$ and an even-sized set of vertices $S \subseteq V_K$, and we need a minimum weight set M of hyperedges such that every T-cut is covered by M; that is, for any $Q \subset V$ with $|Q \cap S|$ odd, there is a hyperedge $e \in M$ intersecting both Q and $V \setminus Q$. Such a set M is called a T-join, and a minimum-weight one can be computed in polynomial-time if K is a graph (Chapter 29 of [26]) . Our generalization of Minimum Weight Graph T-join, which we call *Hypergraph T-join*, is however NP-hard (Section 2) and we cannot directly use the Christofides approach; instead we resort to Robins-Zelikovsky. For our new lower bound, we also need to know the maximum, over two-edge-connected instances of Hypergraph T-join, of the minimum weight T-join divided by the weight of (all edges in) the hypergraph. For graphs, this *T-ratio* is known (and not too hard to prove) to be 1/2 [9]. For hypergraphs, we discovered a sequence of examples where the T-ratio converges to 2/3, and prove it is at most 7/8 for the particular hypergraphs we need.

Min-Power Strong Connectivity has a flavor similar to two more fundamental problems for which the best known ratio is 2: Min-Cost Strong Connectivity and Min-Cost Two Edge Connectivity, and we hope this extended abstract will renew the interest in those two problems.

2 Preliminaries

In directed graphs, we use *arc* to denote a directed edge. In a directed graph K, an *incoming arborescence* rooted at $x \in V(K)$ is a subgraph T of K such that the underlying undirected graph of T is a tree and every vertex of T other than x has exactly one outgoing arc in T. The vertices of T with no incoming arcs in T are called *leafs*.

Given an arc xy, its undirected version is the undirected edge with endpoints x and y. Arcs xy and yx are antiparallel, and the antiparallel arcs resulting from undirected edge uv are uv and vu; if undirected edge uv has cost then each of the two antiparallel arcs resulting from undirected edge uv have this cost.

An alternative definition of our problem is: we are given a simple undirected graph $G = (V, E)$ and a cost function $c : E \to R_+$. A power assignment is a function $p : V \to R_+$, and it induces a simple directed graph $H(p)$ on vertex set V given by xy being an arc of $H(p)$ if and only if $\{x, y\} \in E$ and $p(x) \geq c(\{x, y\})$. The problem is to minimize $\sum_{u \in V} p(u)$ subject to $H(p)$ being strongly connected. To see the equivalence of the definition, given directed spanning subgraph H, define for each $u \in V$ the power assignment $p(u) = p_H(u)$.

The following example shows that the ratio of 2 for the MST algorithm is tight. Consider $2n$ points located on a single line such that the distance between consecutive points alternates between 1 and $\epsilon < 1$, and c be the square of the

Euclidean distance (see Figure 1). Then the minimum spanning tree MST connects consecutive neighbors and has power $p(MST) = 2n$. On the other hand, the tree T with edges connecting each other node (see Figure 1(b)) has power equal $p(T) = n(1+\epsilon)^2 + (n-1)\epsilon^2 + 1$. When $n \to \infty$ and $\epsilon \to 0$, we obtain that $p(MST)/p(T) \to 2$.

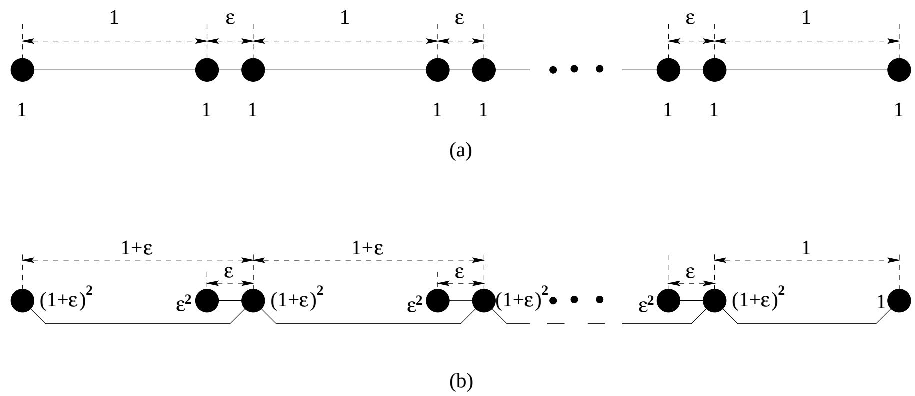

Fig. 1. Tight example for the performance ratio of the MST algorithm. (a) The MST-based power assignment (b) Optimum power assignment

The example above may give intuition on how Power Assignment (and even more specifically, Min-Power Symmetric Connectivity) relates to the k-restricted Steiner trees, with stars (trees of height 1) taking the place of restricted components. Another example from [1], below, shows how Min-Power Strong Connectivity differs from Min-Power Symmetric Connectivity, and may give intuition how Min-Power Strong Connectivity relates to Travelling Salesman, and also Min-Cost Strong Connectivity and Min-Cost Two-Edge Connectivity (a two edge connected graph has an edge orientation that makes it strongly connected - see for example Chapter 2, written by A. Frank, of [12]). However we cannot think of direct reductions either way, and, as we mention in conclusions, the methods we use (more precisely, the Christofides algorithm) only applies to certain instances of Min-Cost Strong Connectivity and Min-Cost Two-Edge Connectivity.

The power of a Min-Power Strong Connectivity optimum solution can be almost half the power of an Min-Power Symmetric Connectivity optimum solution for the same instance: we present a series of examples illustrated in Figure 2. The $n(n+1)$ vertices are embedded in the plane in n groups of $n+1$ points each. Each group has two "terminals" (represented as thick circles in Figure 2), and the $2n$ terminals are the corners of a regular $2n$-gon with sides of length 1. Each group has another $n-1$ equally spaced points (dashes in Figure 2) on the line segment between the two terminals. The cost function c is the square of the Euclidean distance. It is easy to see that a minimum power assignment ensuring strong connectivity assigns a power of 1 to one thick terminal in each group and a power of $\epsilon^2 = (1/n)^2$ to all other points in the group - the arcs going clockwise. The total power then equals $n + 1$. For symmetric connectivity it is necessary to assign power of 1 to all but two of the thick points, and of ϵ^2 to the remaining points, which results in total power of $2n - 1 - 1/n + 2/n^2$. Also, keep in mind that the minimum spanning tree solution is a symmetric solution.

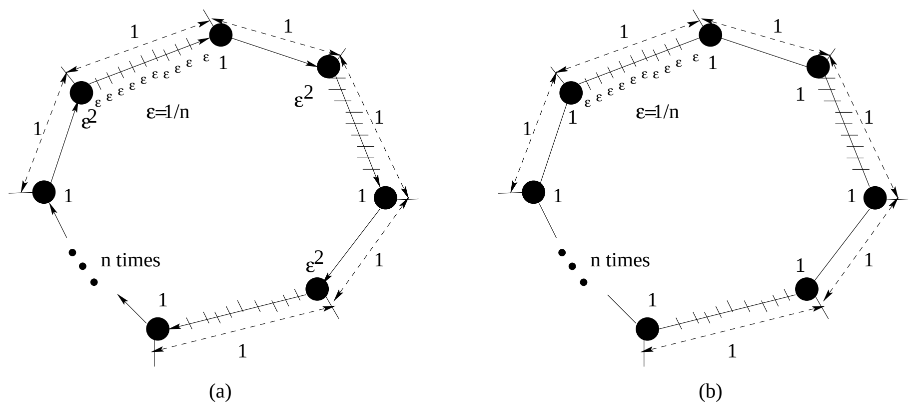

Fig. 2. Total power for the Min-Power Strong Connectivity can be half the total power for Min-Power Symmetric Connectivity. (a) Minimum power assignment ensuring strong connectivity. (b) Minimum power assignment ensuring symmetric connectivity.

While it may be already known, it is easy to check that Hypergraph T-join is indeed NP-hard, by a reduction from 4-D Matching (which asks if a 4-regular hypergraph K with $V(K)$ a multiple of 4, contains a perfect matching, that is, a set of disjoint hyperedges containing every vertex of the input hypergraph; see Garey and Johnson [11] problem SP1). It is easy to check that for $S = V(K)$, a T-join of size at most $|V(K)|/4$ must be a perfect matching.

3 (Weighted) Min-Power Strong Connectivity

Theorem 1. *There exists an $\epsilon > 0$ and a polynomial time algorithm for Min-Power Strong Connectivity with approximation ratio $2 - \epsilon$.*

Our algorithm uses a greedy approach similar to [30, 24, 4]. Let T be the undirected minimum spanning tree of G. We will not use that T is minimum except to note $opt \geq c(T)$, where $opt = p(OPT)$ for an optimum solution OPT. Let $\hat{T}$ be the bidirected version of T.

For $u \in V$ and $r \in \{c(uv) \mid uv \in E\}$, let $S(u, r)$ be the directed star with center u containing all the arcs uv with $c(uv) \leq r$; note that r is the power of S. For a directed star S, let $E(S)$ be its set of arcs and $V(S)$ be its set of vertices.

For given $S(u,r)$, let $Q(u,r)$ be the set of edges of T on a path in T between some $x,y \in V(S(u,r))$. Let $\hat{Q}(u,r)$ be the set of arcs of $\hat{T}$ on a directed path from u to some $x \in V(S(u,r))$; it is easy to verify that the undirected version of $\hat{Q}(u,r)$ is $Q(u,r)$.

For a collection $\mathcal{A}$ of directed stars $S(u_i,r_i)$, define $Q(\mathcal{A}) = \cup_{S(u_i,r_i)\in\mathcal{A}} Q(u_i,r_i)$ and $f(\mathcal{A}) = \sum_{e\in Q(\mathcal{A})} c(e)$. We will use later that this function is increasing and submodular, a well-known fact which appears as an example in Subchapter 44.1.a of [26]. Also define $w(\mathcal{A}) = \sum_{S(u_1,r_i)\in\mathcal{A}} r_i$, the total power used by the stars in $\mathcal{A}$. For $S = S(u,r)$, define $f_{\mathcal{A}}(S) = f(\mathcal{A}\cup\{S\}) - f(\mathcal{A}) = \sum_{e\in Q(u,r)\setminus Q(\mathcal{A})} c(e) = \sum_{e\in I_{\mathcal{A}}(S)} c(e)$, where $I_{\mathcal{A}}(S)$ is defined to be those arcs of $\hat{Q}(u,r)$ for which the undirected version is not in $Q(\mathcal{A})$.

The algorithm starts with $M = \hat{T}$ as the set of arcs, and greedily adds directed stars to collection $\mathcal{A}$ (initially empty) while removing arcs from M to improve the following quantity which is an upper bound on our output: the cost the arcs in M plus the total power of the stars in $\mathcal{A}$. To simplify later proofs, we make changes even if our quantity stays the same. To be precise:

Algorithm **Greedy:**
  $\mathcal{A} \leftarrow \emptyset$, $M \leftarrow \hat{T}$
  While ($f(\mathcal{A}) < c(T)$) do
    $(u,r) \leftarrow \text{argmax}_{(u',r')} f_{\mathcal{A}}(S(u',r'))/r'$
    $M \leftarrow M \setminus I_{\mathcal{A}}(S(u,r))$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{S(u,r)\}$
  Output $\cup_{S\in\mathcal{A}} E(S) \cup M$

Note that a star $S(u,r)$ always exists for which $f_{\mathcal{A}}(S(u,r)) > 0$ and $f_{\mathcal{A}}(S(u,r))/r \geq 1$. Indeed, as long as a pair of antiparallel arcs e' and e'' are in M, we can pick as next star $S(u,r)$ the one given by u being the tail of e' and $r = c(e')$; this star will be added to $\mathcal{A}$ while e' is removed from M.

Lemma 1. *The output is a spanning strongly connected subgraph of G.*

Proof Sketch. We prove the following invariant: $X := \cup_{S\in\mathcal{A}} E(S) \cup M$ gives a spanning strongly connected subgraph whenever the **while** condition is checked by the algorithm. Moreover, suppose we remove from T all edges e for which both antiparallel arcs appear in M, splitting T in components with vertex sets T_j, for some range of j. We prove that for every j and every $x,y \in T_j$, there exists a directed path P from x to y using only vertices of T_j and arcs from X.

The invariant is true before the first iteration, when each T_j has just one vertex, and keeping it is rather straightforward, based on the following idea: when we add an arc uz and remove arcs x_iy_i on the simple path in T from u to z, then arcs y_ix_i are kept and can be used together with paths inside components T_j as above, and arc uz, to produce a directed cycle C containing none of the arcs x_iy_i and passing through u,z and all the y_i,x_i. ■

Lemma 2. *There exists a collection of stars $\mathcal{B}$ with $f(\mathcal{B}) = c(T)$ and $w(\mathcal{B}) \leq (7/8)opt$, where opt is the power of the optimum solution.*

The factor of 7/8 above cannot be improved to a constant better than 1/2, as we can see by looking at the example in Figure 1. To "cover" the whole minimum spanning tree (that is, have $f(\mathcal{B}) = c(T)$) one needs to select every second star of optimum (that is, use with power $(1+\epsilon)^2$ nodes $3, 7, 11 \ldots$). Our proof is based on T-joins and the 7/8 above cannot be improved to more than 2/3 with the same method. Due to page limitations, we omit the long, technical part of the proof.

Proof Idea. Let $(S_v)_{v \in V}$, be the directed stars of OPT, with S_v centered at v, and let $\mathcal{A}$ be collection of these stars. Let $K = (V_K, E_K)$ be the (undirected) hypergraph defined by $V_K = V$ and $E_K = \{V(S) \mid S \in \mathcal{A}\}$. Define the weight of an hyperedge to be the power of the corresponding directed star. A path in a hypergraph consists of an alternating sequence of vertices and hyperedges for which each hyperedge contains the two vertices which precede and follow it in the sequence. A hypergraph is two-edge-connected if there exist two hyperedge-disjoint paths between any two vertices.

Recall from the introduction that, with given $R \subseteq V$ with $|R|$ even, a T-cut is a partition of V into two parts Q and $\bar{Q} := V \setminus Q$ such that $|Q \cap R|$ is odd. A T-join in K for R is a set of hyperedges $M \subseteq E_K$ such that for every T-cut $(Q, \bar{Q})$ has a hyperedge $e \in M$ intersecting both Q and $\bar{Q}$.

The following is the equivalent of Christofides' method:

Claim. Let R be the vertices of the tree T of odd degree (note that $|R|$ is even). Let $\mathcal{D}$ be a set of stars such that the corresponding hyperedges form a T-join in K for R. Then $f(\mathcal{D}) = c(T)$.

Proof. Note that when it comes to computing f, for each star S only $V(S)$ counts, and which vertex is the center is not relevant. For star $S = S(u, r)$, let $Q(S) := Q(u, r)$, defined previously as be the set of edges of T on a path in T between some $x, y \in V(S)$.

We need to show that for every $e \in T$, there is a star $S \in \mathcal{D}$ with $e \in Q(S)$. Indeed, if we remove e from T, we create two subtrees T_u and T_v, where u and v are the endpoints of e. Then $|R \cap V(T_u)|$ is odd, since if we take T_u and add the vertex v and the edge uv, we have an even number of vertices of odd degree, of which one is v. Thus $(V(T_u), V(T_v))$ is a T-cut for R and the T-join given by $\mathcal{D}$ must have an hyperedge intersecting both $V(T_u)$ and $V(T_v)$, and thus a star $S \in \mathcal{D}$ with $e \in Q(S)$. ■

Based on the claim above, Lemma 2 follows from:

Claim. For any arbitrary R, there exists a T-join in K with weight at most $(7/8)w(K)$.

Due to page limitations, we must omit the very long and technical proof of the above claim. We discuss it nevertheless. While K is indeed two-edge-connected (which we do not prove here) and it may very well be the case that the 7/8 ratio holds for any two-edge-connected hypergraph, our current proof relies on the structure of K (how it was obtained from OPT).

The series of example showing that 2/3 is the best we can hope instead of 7/8 also requires too much space. As mentioned before, if we were dealing with graphs rather than hypergraphs the ratio would be 1/2. We only have space for a 3/5 example: nine vertices $x, y_0, y_1, y_2, z_0, z_1, z_2, u_1, u_2$, edges of cost 2: xy_0 and xz_0, edges of cost 1: y_1y_2, z_1z_2, and u_1u_2, and edges of cost 0: y_0y_1, z_0z_1, y_2u_1, z_2u_1, and u_2x. OPT has cost 5: x has power 2 and y_1, z_1, and u_1 each have power 1. The minimum spanning tree has the four edges of cost 0 and the three edges of cost 1. R consists of the vertices x, y_0, z_0, u_1, and one can check by inspection that any T-join has weight 3 (here one can "cover" the minimum spanning tree with just the star of power 2 rooted at x, but our proof method relies on T-joins!). ■

Now we need the following lemma, whose proof is obtained from Robins-Zelikovsky as presented in [13] by changing what quantities represent and some parameters.

Lemma 3. *Assuming that there exists a collection of stars $\mathcal{B}$ with $f(\mathcal{B}) = c(T)$ and $w(\mathcal{B}) \leq \alpha opt$, where opt is the power of the optimum solution and $\alpha < 1$, the algorithms' output has power at most β where $\beta = 1 + \alpha + \alpha \ln(1/\alpha)$.*

Proof. First, if $c(T) \leq \alpha opt$, then before any improvement we have a solution of cost at most $2\alpha opt$ and $2\alpha < \beta$. Thus in the following we assume $c(T) > \alpha opt$.

Note that at the end of the algorithm, M contains one of the two antiparallel arcs for each edge of T. Then the power of the output cannot exceed

$$c(T) + w(\mathcal{A}) \tag{1}$$

for the final $\mathcal{A}$, as a $f(\mathcal{A}) = c(T)$, and for every H and u, $p_H(u) = max_{uv \in H} c(uv) \leq \sum_{uv \in H} c(uv)$.

Let $S_1, S_2, \ldots, S_q$ be the stars picked by our algorithm and let $\mathcal{A}_i$, for $1 \leq i \leq q$ be the collection of the first i stars; also let for convenience $\mathcal{A}_0$ be the empty collection. For $1 \leq i \leq q$, let $p_i = p(S_i) = r_i$, where r_i comes from $S_i = S(u_i, r_i)$, and let $f_i = f_{\mathcal{A}_{i-1}}(S_i)$.

The greedy choice of the algorithm and the submodularity of f gives:

$$\frac{p_i}{f_i} \leq \frac{w(\mathcal{B})}{f_{\mathcal{A}_{i-1}}(\mathcal{B})} = \frac{w(B)}{c(T) - f(\mathcal{A}_{i-1})} = \frac{w(B)}{c(T) - \sum_{j=1}^{i-1} f_j}.$$

Rewriting and replacing $w(\mathcal{B})$ with αopt, we obtain

$$p_i \leq f_i \frac{\alpha opt}{c(T) - \sum_{j=1}^{i-1} f_j}. \tag{2}$$

Define the function $g : [0..c(T)] \rightarrow [0..1]$ by $g(x) = \alpha opt/(c(T) - x)$ for $x \leq c(T) - \alpha opt$, and $g(x) = 1$ for $x > c(T) - \alpha opt$. Then from Equation 2 and the observation (made right after the algorithm) that $\frac{p_i}{f_i} \leq 1$, we obtain:

$$\sum_{i=1}^{q} p_i \le \int_0^{c(T)} g(x)dx = \int_0^{c(T)-\alpha opt} \frac{\alpha opt}{c(T)-x} dx + \int_{c(T)-\alpha opt}^{c(T)} 1 dx$$
$$= (-\alpha opt) \ln(c(T)-x)|_0^{c(T)-\alpha opt} + (c(T) - (c(T) - \alpha opt))$$
$$= \alpha opt \left(1 + \ln \frac{c(T)}{\alpha opt}\right)$$

Using this and $c(T) \le opt$ and Equation 1 (recall that $w(\mathcal{A}) = \sum_{i=1}^{q} p_i$), we obtain that the power of the output is at most

$$c(T) + \alpha opt \left(1 + \ln \frac{c(T)}{\alpha opt}\right) \le opt\,(1 + \alpha + \alpha \ln(1/\alpha))$$

finishing the proof. ■

Based on Lemmas 1 and 2, Theorem 1 follows immediately from the fact that $\alpha < 1$ implies $\beta < 2$, which follows from $\alpha(1 + \ln(1/\alpha)) < 1$, which is equivalent to $\ln(1/\alpha) < 1/\alpha - 1$, a fact that holds for all $\alpha > 1$. For $\alpha = 7/8$, we obtain $\beta \le 1.992$. If one were to prove $\alpha = 1/2$ or $\alpha = 2/3$ in Lemma 2, the resulting approximation ratio would be less than 1.85 or 1.94 respectively.

4 Min-Power Strong Connectivity with two power levels

In this section, it is more convenient to to work with the alternative definition of power assignment described in the preliminary section.

Theorem 2. *For every integer $k \ge 4$, there exists a $O(n^{k+3})$ algorithm with approximation ratio $\pi^2/6 - 1/36 + 1/(k(k-1))$.*

Given set of vertices $S \subseteq V$, we define the function $p^S : V \to \{0, 1\}$ as follows: $p^S(u) = 1$ if $u \in S$ and $p^S(u) = 0$ if $u \in V \setminus S$.

We say that $S \subseteq V$ is *impeccable* if no arc of $H(p^S)$ has endpoints in distinct strongly connected components of $H(p^S)$. See Figures 3 and 4 for illustrations, where we use the following conventions: In our pictures, we use full small circles to denote nodes with power 1, that is, S. Empty little circles are nodes with power 0, continuous thin line segments are edges of E of cost $c = 0$; both antiparallel corresponding arcs are in H. Thicker segments with arrows are for arcs of $H(p^S)$ coming from edges of E of cost 1, and thicker dashed segments for edges of E of cost 1 for which none of the two antiparallel corresponding arcs are in H. Larger ellipses represent strongly connected components of $H(p^S)$.

We say that $Q \subseteq V$ is *quasi-perfect* with respect to impeccable $S \subseteq V$ if the following holds: Let $X_1, X_2, \ldots, X_s$ be the strongly connected components of $H(p^S)$. Then for all $i = 1, 2, \ldots, q$, $|Q \cap X_i| \le 1$, and all vertices of Q are in the same strongly connected component of $H(p^{Q \cup S})$. See Figure 5 for illustrations. If Q is quasi-perfect w.r.t impeccable S and $Q \cup S$ is impeccable, then Q is called *perfect* w.r.t. S.

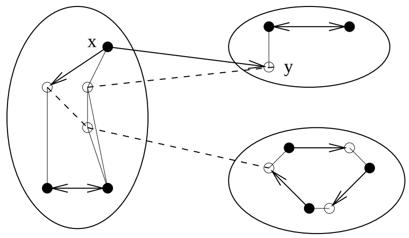

Fig. 3. This S is not impeccable as $H(p^S)$ has the arc xy crossing between its strongly connected component.

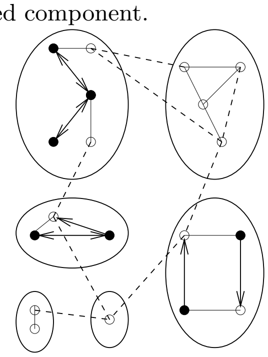

Fig. 4. An impeccable S.

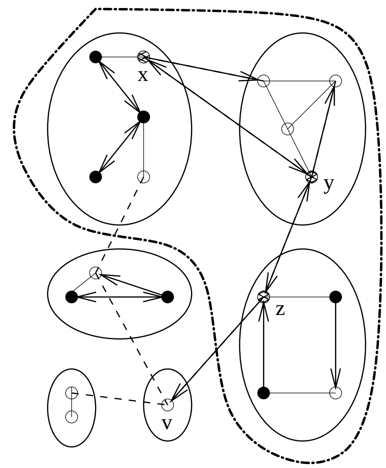

Fig. 5. The set $Q = \{x, y, z\}$ is quasiperfect w.r.t the given S as the thick dotted closed curve gives the strongly connected component of $H(p^{Q \cup S})$ containing Q. Q is not perfect since the arc zv has endpoints in different strongly connected component of $H(p^{Q \cup S})$.

Our algorithm grows an impeccable set of vertices S (S_i, below) by adding to S a perfect set Q. Perfect sets of vertices play for us the role of directed cycles in the Khuller et al. algorithms [17, 18]. In [17, 18] strongly connected components are contracted, for simplicity, to a single vertex; we have to keep them during the algorithm. Below is our modified version of the Khuller et al. algorithm. Both running time and approximation ratio depend on integer parameter $k \geq 4$.

1 Set $i = 0$, $j = k$, $S_0 = \emptyset$.
2 **while** $j \geq 4$
3   **if** there exists $Q \subseteq V$ with $|Q| \geq j$ and perfect w.r.t. S_i
4       $S_{i+1} = S_i \cup Q$; $\quad i = i + 1$
5   **else**
6       $j = j - 1$
7 **endwhile**
8 Find minimum $R \subseteq V$ that makes $H(p^{S_i \cup R})$ strongly connected
9 Return $p^{S_i \cup R}$

We need to elaborate for step 3 and step 8, which differ from [17, 18]. Step 3 adapts the [17, 18] method; this cannot be done for Step 8. Due to space limitations we omit our solution (based on Graphical Matroid Parity [21, 10, 27]), but mention that we believe the algorithms of Carmi and Katz [5] gives the optimum indeed for instances where every perfect set has at most three vertices.

Step 3 First, Step 3 can be implemented in $O(n^{k+2})$, as described in the following procedure, whose input is S_i (from now on we use S instead of S_i). Compute $X_1, X_2, \ldots, X_s$, the strongly connected components of $H(p^S)$. Try all combinations of sets $J \subseteq S \setminus S_i$ of exactly j elements. If there exists a $k \in \{1, 2, \ldots, s\}$ with $|J \cap X_k| > 1$, reject J (and try another combination).

Otherwise, construct the directed graph $L = L(S, J)$ with vertex set $\{k \mid |J \cap X_k| = 1\}$ and put an arc from k to q if there exists an edge $e \in E$ endpoints in $J \cap X_k$ and X_q. See Figure 6 for an illustration of this paragraph. Compute the strongly connected components of L. If L is strongly connected proceed to the extension phase, described in the next paragraph. If L has more than one strongly connected component with no incoming arc, or more than one strongly connected component with no outgoing arc, reject J. Else, let Y be the vertex set of the strongly connected component of L with no incoming arc, and Z be the vertex set of the strongly connected component of L with no outgoing arc. Construct an undirected simple graph M with vertex set $\{v_k \mid |J \cap X_k| = 0 \text{ or } k \in Y\}$ and put an edge from v_k to v_q if there exists an edge $e \in E$ with endpoints in X_q and X_k. Let $W = \{v_k \in V(M) \mid \exists x \in J \cap (\cup_{r \in Z} X_r)\ \exists y \in X_k\ \exists e \in E\ (e = \{x, y\})\}$. If M does not have a path from some v_k with $k \in W$ to some v_q with $q \in Y$, then reject J. Else, let P be such a path with minimum number of edges. Each edge f of P comes from an edge $\{x_f, y_f\} \in E$, and permute x_f and y_f if necessary such that, if $x_f \in X_k$ and $y_f \in X_q$, then k is before q on P when starting from W. Put all such x_f in J and proceed.

Now we have the extension phase: as long as there exists an edge $e \in E$ with one endpoint in J and the other in some X_k with $|J \cap X_k| = 0$, we add to J the endpoint of e in X_k.

Due to space limitations, the correctness and analysis of this algorithm is left for a full (journal) version of this extended abstract.

5 Conclusions

Based on the discussion in Section 3, we also have: if we are looking for a minimum weight strongly connected spanning subgraph in a directed graph, and a bidirected tree exists of cost at most twice optimum, then Christofides' method gives a 1.5 approximation. The same holds for Minimum Weight Two-Edge-Connected Spanning Subgraph if there always exists a "double tree" (that is, where each edge has a twin) of weight at most twice optimum. These facts are probably known.

The approximation ratio of Khuller, Raghavachari, and Young [18] has been improved to 1.5 by Vetta [28]. However we see no way of adapting [28] (or [16]) to power assignment with two power levels: the first step of [28] (Lemma 3.1) is to split the graph until no cut vertex exists. For us, the equivalent of a cut

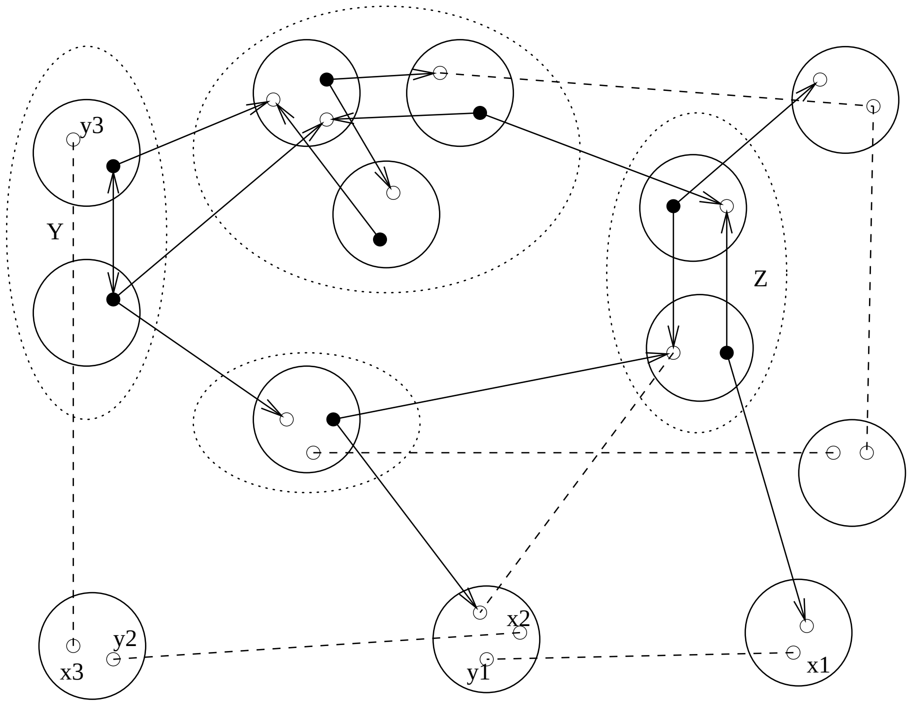

Fig. 6. Here circles are the strongly connected components of $H(p^S)$; members of S are not depicted and the filled small circles give the set J. Empty small circles are in neither S nor J. The dashed ellipses give the strongly connected components of L (we kept parallel arcs in the picture). In this example, Y and Z exist and are depicted. M is also implicit in the figure: the solid circles not in dashed ellipse, and here one such path P can be x_1y_1, x_2y_2, and x_3y_3.

vertex would be a strongly connected component of $p^{\emptyset}$ - and what we choose in this strongly connected component can affect all the graph as in Figure 7. Thus we cannot split the input graph, and with cut vertices the lower bound of [28] is not 1.5 but 2.

It is possible that the very recent breakthrough by Byrka et. al [3] for Steiner Tree gives techniques to improve our approximation ratio for Min-Power Strong Connectivity.

We leave open the correct T-ratio in two-edge-connected hypergraphs.

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. Christoph Ambühl, Andrea E. F. Clementi, Paolo Penna, Gianluca Rossi, and Riccardo Silvestri. On the approximability of the range assignment problem on

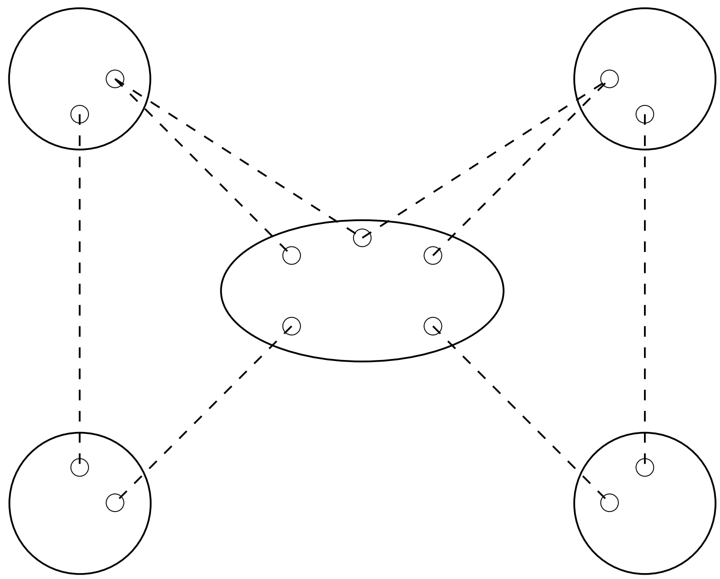

Fig. 7. Splitting the graph in two by the middle cut-vertex results in a loss in approximation ratio as some vertices of G - like the middle one here - can contribute to both sides of the split.

radio networks in presence of selfish agents. *Theor. Comput. Sci.*, 343(1-2):27–41, 2005.

3. Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. An improved LP-based approximation for Steiner tree. In *STOC '10: Proceedings of the 42nd ACM symposium on Theory of computing*, pages 583–592, New York, NY, USA, 2010. ACM.
4. Ioannis Caragiannis, Michele Flammini, and Luca Moscardelli. An Exponential Improvement on the MST Heuristic for Minimum Energy Broadcasting in Ad Hoc Wireless Networks. In *ICALP*, pages 447–458, 2007.
5. Paz Carmi and Matthew J. Katz. Power assignment in radio networks with two power levels. *Algorithmica*, 47(2):183–201, 2007.
6. W.T. Chen and N.F. Huang. The strongly connecting problem on multihop packet radio networks. *IEEE Transactions on Communications*, 37(3):293–295, 1989.
7. N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Carnegie Mellon University, 1976.
8. A.E.F. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. *Electronic Colloquium on Computational Complexity (ECCC)*, (054), 2000.
9. Jack Edmonds and Ellis L. Johnson. Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5(1):88–124, 1973.
10. H.N. Gabow and M. Stallmann. Efficient algorithms for graphic matroid intersection and parity. In *12th Colloq. on Automata, Language and Programming*, pages 210–220, 1985.
11. M. R. Garey and D. S. Johnson. *Computers and Intractability*. W.H. Freeman and Co., NY, 1979.

12. R.L. Graham, M. Grötschel, and L. Lovász, editors. *Handbook of combinatorics. Vol. 1, 2*. Elsevier Science B.V., Amsterdam, 1995.
13. C. Gröpl, S. Hougardy, T. Nierhoff, and H.J. Prömel. Approximation algorithms for the Steiner tree problem in graphs. In D.-Z. Du and X. Cheng, editors, *Steiner Trees in Industries*, pages 235–279. Kluwer Academic Publishers, Dordrecht, 2001.
14. Mohammad Taghi Hajiaghayi, Nicole Immorlica, and Vahab S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In *MOBICOM*, pages 300–312, 2003.
15. Mohammad Taghi Hajiaghayi, Guy Kortsarz, Vahab S. Mirrokni, and Zeev Nutov. Power optimization for connectivity problems. *Math. Program.*, 110(1):195–208, 2007.
16. Raja Jothi, Balaji Raghavachari, and Subramanian Varadarajan. A 5/4-approximation algorithm for minimum 2-edge-connectivity. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 725–734, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.
17. Samir Khuller, Balaji Raghavachari, and Neal Young. Approximating the minimum equivalent digraph. *SIAM J. Comput.*, 24(4):859–872, 1995.
18. Samir Khuller, Balaji Raghavachari, and Neal Young. On strongly connected digraphs with bounded cycle length. *Discrete Appl. Math.*, 69(3):281–289, 1996.
19. L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243:289–305, 2000.
20. Guy Kortsarz, Vahab S. Mirrokni, Zeev Nutov, and Elena Tsanko. Approximating minimum-power degree and connectivity problems. In *LATIN*, pages 423–435, 2008.
21. L. Lovász and M.D. Plummer. *Matching Theory*. Elsevier, 1986.
22. Zeev Nutov and Ariel Yaroshevitch. Wireless network design via 3-decompositions. *Inf. Process. Lett.*, 109(19):1136–1140, 2009.
23. Ram Ramanathan and Regina Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM (2)*, pages 404–413, 2000.
24. G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 770–779, 2000.
25. G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM Journal of Discrete Mathematics*, 19(1):122–134, 2005.
26. A. Schrijver. *Combinatorial Optimization*. Springer, 2003.
27. Z. Szigeti. On the graphic matroid parity problem. *J. Combin. Theory Ser. B*, 88:247–260, 2003.
28. Adrian Vetta. Approximating the minimum strongly connected subgraph via a matching lower bound. In *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 417–426, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.
29. Roger Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang. Distributed topology control for wireless multihop ad-hoc networks. In *IEEE INFOCOM'01*, 2001.
30. A. Zelikovsky. Better approximation bounds for the network and Euclidean Steiner tree problems. Technical Report CS-96-06, Department of Computer Science, University of Virginia, 1996.