J Comb Optim (2010) 19: 174–183
DOI 10.1007/s10878-008-9173-x

On dual power assignment optimization for biconnectivity

Chen Wang · James Willson · Myung-Ah Park · Andras Farago · Weili Wu

Published online: 1 August 2008

Abstract Topology control is an important technology of wireless ad hoc networks to achieve energy efficiency and fault tolerance. In this paper, we study the *dual power assignment problem* for 2-edge connectivity and 2-vertex connectivity in the symmetric graphical model which is a combinatorial optimization problem from topology control technology.

The problem is arisen from the following origin. In a wireless ad hoc network where each node can switch its transmission power between high-level and low-level, how can we establish a fault-tolerantly connected network topology in the most energy-efficient way? Specifically, the objective is to minimize the number of nodes assigned with high power and yet achieve 2-edge connectivity or 2-vertex connectivity.

We addressed these optimization problems (2-edge connectivity and 2-vertex connectivity version) under the general graph model in (Wang et al. in Theor. Comput. Sci., 2008). In this paper, we propose a novel approximation algorithm, called Candidate Set Filtering algorithm, to compute nearly-optimal solutions. Specifically, our algorithm can achieve 3.67-approximation ratio for both 2-edge connectivity and 2-vertex connectivity, which improves the existing 4-approximation algorithms for these two cases.

Keywords Dual power assignment · Biconnectivity · Approximation algorithm

1 Introduction

Mobile Ad hoc Network (MANET) has attracted significant attention due to a broad range of applications in environmental monitoring, military operation and health applications, and also its challenging research problems. Since a mobile device usually

C. Wang (✉) · J. Willson · M.-A. Park · A. Farago · W. Wu
Tsinghua University, Room 4-609, FIT Bldg., Beijing, 100084, China
e-mail: wsamc82@gmail.com

is battery-powered and vulnerable, energy efficiency and fault tolerance are two main issues in a wireless ad hoc network.

Topology control is to achieve the conflicting goals of saving energy while maintaining fault tolerance. The basic framework of topology control is to adjust transmission power at each node in a network according to desired features of the network such as connectivity, 2-connectivity and low interference among neighbors. A transmission power from a node u to another node v is proportionate to $d(u,v)^c$ in most used radio propagation models, where $d(u,v)$ is the Euclidean distance between u and v and c is the power attenuation exponent, typically between 2 and 4. So, we assume that for a given transmission power p, there exists a unique corresponding transmission range r. In other words, we regard a power assignment as a transmission range assignment.

A topology or communication graph induced by power adjustment is usually a directed graph due to the asymmetricity of transmission powers among neighbor nodes. However, it has been shown in the literature that a unidirectional link is detrimental to the performance of a network. So, it is necessary to symmetritize edges among neighbor nodes. Here, we assume that there exists a (*symmetric*) *link* between nodes u and v if $d(u,v) \le r_u$ and $d(u,v) \le r_v$ where r_u and r_v are the transmission ranges of u and v, respectively. In other words, we prune all unidirectional links induced by a power assignment to obtain a symmetric topology.

Extensive studies has been done on topology control in the literature. Most works assume a *continuous* power assignment: a node can set its transmission range to r where r can be any value in $[r_{min}, r_{max}]$. A topology control algorithm based on continuous power assignment uses the triangle inequality to derive the concept of energy efficiency. In reality, however, each sensor is given k different transmission powers where k is a small constant. In this paper, we consider the simplest case that there are only two universal transmission power levels. With dual power assignment, minimum number of maximum-power nodes and the minimum total power, which is defined as the sum of the power of each node, are achieved at the same time. However, these two optimization goals are different with respect to approximation ratio. It is easy to see that an α-approximation for the minimum number of high-power nodes is always an α-approximation for the minimum total power, but the inverse is not true. It is also shown in Poojary et al. (2001) and Shah et al. (2004) that it is important to minimize the number of maximum-power nodes, when a *discrete* power assignment is considered. If the objective of a discrete power assignment is to minimize the number of maximum-power nodes, we can assume without loss of generality that there are two transmission power levels available.

For a given set of nodes V in a Euclidean plane, a dual power assignment is defined as a range assignment A where $A(v) = r_h$ or $A(v) = r_l$ where r_h and r_l are the high and the low transmission ranges, respectively. The *low-power graph* G_l is the graph induced by the low-power assignment, i.e., $A(v) = r_l$ for all $v \in V$, and the *high-power graph* G_h is the graph induced by the high-power assignment, i.e., $A(v) = r_h$ for all $v \in V$. The goal is to find an assignment which minimize the number of high power nodes and meanwhile induce an 2-edge (2-vertex) connected graph. The definition above can also be generalized in a general undirected graph model as follows.

Definition 1 (Dual power assignment for 2-edge (vertex) connectivity problem) Given a set of nodes V, the 2-edge(vertex) connected graph $G_h = (V, E_h)$ and its subgraph $G_l = (V, E_l)$ where $E_l \subseteq E_h$, find an edge set $E \subseteq E_h \setminus E_l$, such that $G = (V, E_l \cup E)$ is 2-edge(vertex) connected, while the size of $V(E)$ is minimized, where $V(E)$ denotes the set of vertices, each of which has at least one attached edge in the set E.

Note that in our definition, graph nodes are no longer limited in a plane. Any network topology which can be modeled as a graph is allowed. We will use DPA-2EC and DPA-2VC to abbreviate the two problems, respectively. In addition, one can prove that the DPA-2EC and DPA-2VC problem are NP-hard by using reduction from the minimum set cover problem.

The rest of the paper is organized as follows. We firstly introduce some related works in Sect. 2. Then, in Sect. 3, some preliminaries and basic intuitions of our algorithm are given. Our approximation algorithms are formally described in Sects. 4 and 5. Specifically, in Sect. 4, we proposed the Candidate Set Filtering Algorithm for the DPA-2EC problem followed by a proof of the approximation results, and in Sect. 5, we achieve the 3.67-approximation for DPA-2VC by using a similar algorithm with a transformation technique. As the last part, we conclude the paper and give some future works in Sect. 6.

2 Related works

The power-optimal continuous range assignment problem has been studied extensively. As this problem is NP-hard even in the Euclidean plane (Clementi et al. 1999), some approximation algorithms are developed. Kirousis et al. (2000) give a 2-approximation algorithm by constructing the minimum spanning tree of the network graph. Calinescu and Wan (2006) improve the approximation factor to 1.69 with a steiner tree based algorithm. As a natural generalization of connectivity, k-connectivity or k-edge connectivity power range assignment problem is also studied in previous works. For the special case of biconnectivity, Lloyd et al. (2002) present a 8-approximation algorithm. For general k-connectivity and k-edge connectivity problem, an $O(k)$-approximation centralized algorithm is designed by Hajiaghayi et al. (2003). Calinescu and Wan (2006) improve the approximation result for k-edge connectivity to $2k$, and for biconnectivity to 4.

As our best known, the dual power assignment problem is first studied by Rong et al. (2004). The authors define the asymmetric version of the problem, prove the NP-hardness of strongly connected dual power assignment problem, and design a 2-approximation algorithm based on graph theoretic facts. Chen et al. (2005) improved the approximation factor of asymmetric version to 1.75. Lloyd and Liu (2006) study the minimum number of maximum power users problem, which is actually the symmetric version DPA problem, prove the NP-hardness of the symmetric version and present a 1.67-approximation algorithm. But none of these works consider fault tolerance.

The first work with fault tolerance taken in account is Park et al. (2006). The authors proposed the minimum spanning tree augmentation algorithms for 2-edge connectivity and 2-vertex connectivity with dual power assignment. The algorithms are quite simple and achieve approximation ratio of 6 for 2-edge connectivity and 5 for 2-vertex connectivity respectively. In Wang et al. (2008), another approximation algorithm is designed to achieve an improved approximation of 4 for both 2-edge connectivity and 2-vertex connectivity.

3 Preliminaries

In this section, we give the definition of 2-edge(vertex) connectivity and list some lemmas which describe some basic properties of 2-edge(vertex) connectivity and will be used for our algorithms. Some of them can be found in textbook of graph theory, e.g., (Diestel 2000) and others can be proved without difficulties. We just list them here without proof due to the page limit.

Definition 2 An undirected graph G is called 2-*edge* (*2-vertex, resp.*) *connected* if, G remains connected after removing any one edge (vertex, resp.).

The following lemmas give some characteristics of 2-edge(vertex) connectivity.[1] We will use them to prove the correctness as well as an approximation ratio of our algorithms in Sects. 4 and 5.

Lemma 1 *Given a* 2*-edge connected graph* $G = (V, E)$ *and a non-trivial subset of nodes*, U, *there are at least two edges between* U *and* $V \backslash U$.

Lemma 2 *Given a* 2*-vertex connected graph* $G = (V, E)$ *and a non-trivial subset of nodes*, U, *where* $|U| > 1$, *there are at least two nodes in* U *connecting to* $V \backslash U$.

Lemma 3 *Given a connected graph* $G = (V, E)$ *and a spanning tree* $T = (V, E_T)$ *of* G, G *is* 2*-edge connected iff for each edge* $e \in E_T$ *of* T, *there is an edge* $e' \in E$, *such that the new graph* $(V, E_T - \{e\} + \{e'\})$ *is still connected.*

Lemma 4 *Given a connected graph* $G = (V, E)$ *and a spanning tree* $T = (V, E_T)$ *of* G, G *is* 2*-vertex connected iff for each vertex* $v \in V$ *of* T, *there are an edge set* $E' \in E - E_T$, *such that the new graph* $(V, E_T - E_T(v) + E')$ *is still connected*, *where* $E_T(v)$ *is the set of all edges in* E_T *attached to* v.

Our algorithm is motivated by Lemmas 3 and 4. The basic idea is to construct a spanning tree of G_h and then add more high power nodes to make it 2-edge or 2-vertex connected.

In the first step, a straightforward method is to compute a minimum weighted spanning tree T based on the following weight assignments: for edge $e \in E_h$, $w(e) =$

[1] Lemmas 2 and 4 are analog for 2-vertex connectivity of Lemmas 1 and 3, respectively.

1 if e is a high power edge, and otherwise set $w(e) = 0$. If we denote U_1 as the number of high power nodes used to construct T, we can see

$$w(T) \triangleq \sum_{e \in T} w(e) \geq \frac{|U_1|}{2}.$$

Moreover, if we assume that G_l has c different connected components, we know

$$w(T) = c - 1.$$

Let us use opt_e and opt_v to denote the optimal solutions of DPA-2EC and DPA-2VC, respectively. According to Lemma 1, it can be shown that $c \leq opt_e$, and $c \leq opt_v$. Consequently we know $U_1 \leq 2 \cdot opt_e$ and $U_1 \leq 2 \cdot opt_v$. In fact, this 2-ratio can be improved to 1.67 by using the greedy algorithm in Lloyd and Liu (2006).

4 Approximation algorithm for DPA-2EC

In this section, we propose an algorithm called *Candidate Sets Filtering Algorithm* (CSFA) to solve the DPA-2EC problem.

The algorithm includes two steps. In the first step, we construct a *rooted* spanning tree T of the network by using the previous algorithm in Lloyd and Liu (2006). In the second step, for each edge in the tree, we will check if there is another low power edge connecting the two separated parts formed by removing the edge from T, otherwise we use a high power edge to connect them. It is sufficient to guarantee the 2-edge connectivity of the resultant graph according to Lemma 3.

Before we give more details, we need introduce some basic terminologies and notations.

- The rooted spanning tree constructed in step 1 is denoted by T. The root is denoted by r.
- The parent of a node v is denoted by $p(v)$.
- A **tree edge** is an edge in the rooted tree T.
- A **GREEN edge** is an edge which is in E_l but not in T, i.e., a low-power non-tree edge.
- A **RED edge** is an edge which is in $E_h \setminus E_l$ but not in T, i.e., a high-power non-tree edge.
- The **rooted subtree** of a node v, denoted by T_v, is the subtree of T rooted at v.
- For a node v other than the root, the **inside** part of v is defined as T_v and the **outside** part of v is defined as $T - T_v$.
- A node v other than r is **covered** by a GREEN or RED edge e if e connects the insider part and outside part of v.
- The **candidate set** of a node v, denoted by $CS(v)$ is a set of RED edges which is created during the algorithm runs. More details can be found in the algorithm description.

Given a rooted tree, the second step computes an 2-edge connected graph. As initialization, for each node other than r, we mark it WHITE and create an empty candidate set. Then the algorithm explores nodes bottom-up along T. For each leaf v, if it has another low-power edge other than $(v, p(v))$, i.e., a GREEN edge, we mark v BLACK to indicates this node is already explored. Otherwise, we will fulfill its candidate set with all of its RED edges and then mark it BLACK.

For each non-leaf WHITE node v, it is explored after all of its children are already BLACK. In this iteration, our purpose is to check if there is either a GREEN edge covering v. If such a GREEN edge exists, we do nothing but color v BLACK. Otherwise, we check whether there is some edge in any inside candidate set covering v for the candidate set of each node in T_v, if it has at least one edge covering v, we "filter" this candidate set by removing any other edges which is an edge inside T_v. If neither such a GREEN edge nor a candidate set exists, we will fulfill $CS(v)$ with all of RED edges covering v. Eventually, we mark v BLACK to indicate it has been processed.

The algorithm runs iteratively on WHITE nodes and the number of WHITE nodes decreases by one in a single iteration. So our algorithm always terminates with an output.

The correctness of the algorithm is implied by the following lemma.

Lemma 5 *CSFA outputs a power assignment which results in a* 2*-edge connected graph.*

Proof It is easy to see that the resulting graph of output high power nodes has all edges in T and exact one edge in each non-empty candidate set.

Since T is connected, according to Lemma 3, we only need to prove that for each tree edge $(v, p(v))$, there is at least another edge in the resulting graph covering v. As we known, if $CS(v)$ is non-empty, all the edges in $CS(v)$ can cover v. Note that $CS(v)$ keeps non-empty even though some of its edges may be removed. On the other hand, if $CS(v)$ is eventually empty, it means that there is either a GREEN edge or an edge in some existing candidate set $CS(v')$ covering $(v, p(v))$. Moreover, if the latter case happens, we will filter $CS(v')$ and remove all edges in it which does not cover v. So this guarantees that there is eventually either a GREEN edge covering v or a candidate set $CS(v')$ where all edges in it covering v. So the lemma holds. □

In addition, we can show that the CSFA has a 3.67-approximation ratio. This result is proved in the following theorem.

Theorem 1 *CSFA computes a* 3.67*-approximation of DPA-*2*EC problem.*

Proof The output set of high power nodes consists of two parts, U_1 and U_2. The approximation result from (Lloyd and Liu 2006) directly implies that $|U_1| \leq 1.67 \cdot opt_e$. It is sufficient to prove $|U_2| \leq 2 \cdot opt_e$. This is done based on Lemma 1.

First of all, as we arbitrarily choose an edge from non-empty candidate set and add the two endpoints into U_2,

$$|U_2| \leq 2 \cdot |\{\text{non-empty candidate sets}\}|.$$

So it is sufficient to prove that the number of non-empty candidate sets is at most opt_e. This follows two observations: 1. a non-empty candidate set $CS(v)$ is created

Algorithm 1 Candidate Sets Filtering Algorithm for DPA-2EC

INPUT: (V, G_h, G_l)
OUTPUT: Power assignment such that the resulting symmetric graph is 2-edge connected.

Step 1.
Construct a spanning tree T of G_h rooted at r by the algorithm in (Lloyd and Liu 2006), and assume the set of used high power nodes is U_1;

Step 2.
for each node v except r **do**
  Mark v WHITE;
  $CS(v) \leftarrow \emptyset$; /* *Initialize all candidate sets* */
end for
while there exists at least a WHITE node **do**
  Find a WHITE node, say v, which has no WHITE children;
  $c \leftarrow 0$;
  for each node $v' \in T_v$ with $CS(v') \neq \emptyset$ **do**
    if there is at least one edge in $CS(v')$ which covers v **then**
      $c \leftarrow 1$;
      for each edge e' in $CS(v')$ whose two endpoints are both in $T_{v'}$ **do**
        $CS(v') \leftarrow CS(v') \backslash \{e'\}$;
      end for
    end if
    if $c = 0$ and there is not a GREEN edge covering v **then**
      $CS(v) \leftarrow CS(v) \cup \{e|e$ is red, and e covers $v\}$;
    end if
    Mark v BLACK;
  end for
end while
$U_2 = \emptyset$; /* *Initialize the output set of Step 2* */
for each v in the graph **do**
  if $|CS(v)| > 0$ **then**
    choose an arbitrary edge, say (u_1, u_2), from $CS(v)$ and let $U_2 \leftarrow U_2 \cup \{u_1, u_2\}$;
  end if
end for
Output $U_1 \cup U_2$.

only if there is no GREEN edge or other edges in some existing candidate set to cover the underlying edge (between v and its parent); 2. assume that v' in T_v has an attached edge added into $CS(v)$ when v is processed, it will not has attached edges added when any ancestors of v is processed.

The first observation ensures that when a non-empty candidate set $CS(v)$ is created, there is at least one high power node inside T_v which is also in OPT_e according to Lemma 1. In addition, such a high power node in OPT_e has at least an attached edge added into $CS(v)$ when v is processed.

The second observation can be shown by contradiction. That is, if v' has at least an edge in $CS(v)$ and also an edge in $CS(u')$ which u' is some ancestor of v, then let us consider the iteration of processing u'. As $CS(u')$ is non-empty, it means $CS(v)$ has no edge to cover u'. However, if v' has an attached edge in $CS(u')$, this edge should also be added into $CS(v)$ and has not been removed when $CS(u')$ is fulfilled. This contradicts that $CS(v)$ has no edge to cover u'.

At last, we define the **candidate node set** of v is the set of all nodes in T_v which has at least one attached edge added into $CS(v)$ in the iteration when v is processed. From two observations, we actually prove the optimal solution OPT_e has at least one high power node for each non-empty candidate node set and two candidate node sets are disjoint. So we see

$$\begin{aligned} |U_2| &\le 2 \cdot |\{\text{non-empty candidate sets}\}| && (1)\\ &= 2 \cdot |\{\text{non-empty candidate node set}\}| && (2)\\ &\le 2 \cdot opt_e. && (3) \end{aligned}$$

From all above, we can conclude our algorithm computes a 3.67-approximation. □

5 Approximation algorithm for DPA-2VC

In this section, we modify the Candidate Set Filtering Algorithm to solve the DPA-2VC problem. To avoid confusion, we will use CSFA-2VC to denote the modified algorithm. All terminologies of CSFA except the following one can be applied immediately:

- A node v other than r is **covered** by a GREEN or RED edge e if e connects $T \setminus T_v$ and T_v, and $p(v)$ is not attached by e.

CSFA-2VC also starts with a rooted spanning tree by the algorithm in Lloyd and Liu (2006). However, the second step need to be modified. Roughly speaking, according to Lemma 4, we need to consider each node of the tree in a bottom-up order. After check if the removal of current node disconnects the graph by looking up existing non-empty candidate sets, we update candidate sets accordingly.

An observation is, for a node v with at most 2 children, if there is at least one edges covering v and at least one covering each of its children respectively, the removal of v cannot disconnected the graph. This will be proved in Lemma 6. Based on the observation, we locally transform a node with more than 2 children to a binary tree, and use the similar process on this subtree as that of CSFA.

In concrete, when we process a node v with k children $w_1, w_2, \ldots, w_k$, the subroutine splits v into $k-1$ nodes $v_1, v_2, \ldots, v_{k-1}$, and construct a binary tree consisting of $v_1, v_2, \ldots, v_{k-1}, T_{w_1}, T_{w_2}, \ldots, T_{w_{k-1}}$, and $T \backslash T_v$. This operation is illustrated in Fig. 1. Note that we require that each v_i has at most two children regardless of other parts of the tree in this iteration. We call the graph obtained from T by splitting v the

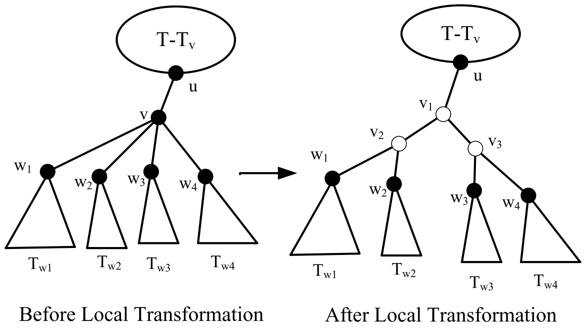

Fig. 1 The local transformation operation for each iteration of CSFA-2VC

local binary tree of v, denoted by BT_v. A subtree of BT_v rooted at node u is denoted by $BT_v(u)$.

After we got BT_v, we consider each v_i as a real WHITE node and check if there is either a GREEN edge or a candidate set which has a edge covering v_i. If so, we filter all the candidate sets of his children in the same way as CSFA. Otherwise, we fulfill $CS(v_i)$ with all edges from $BT_v(v_i) \setminus \{v_i\}$ to any of v_i's ancestors. At the end of the iteration when v is processed, we mark v BLACK in T.

Note that in CSFA, if node v is splitted into k nodes in its iteration, all $CS(v_k)$ are regarded as independent candidate sets in following iterations where v's ancestor is processed. Eventually, after all WHITE nodes are done, we choose an edge from each candidate set and output the set of high power nodes accordingly.

The correctness of CSFA-2VC is proved in the following lemma.

Lemma 6 *The CSFA-2VC outputs a power assignment which results in a 2-vertex connected graph.*

Proof According to Lemma 4, we need only check that, for each node v in T, all subtrees rooted at v's children and $T \setminus T_v$ are connected. We prove this by induction.

As induction basis, a single leaf apparently satisfies this property. Then, we assume that all children of a node v have satisfied this property. Here we need consider two cases. In the first case that v has at most 2 children, say u_1 and u_2. It implies the removal of v divides T into 3 parts: T_{u_1}, T_{u_2} and $T \setminus T_v$. According to induction hypothesis, T_{u_1} is connected to either T_{u_2} or $T \setminus T_v$, and T_{u_2} is connected to either T_{u_1} or $T \setminus T_v$ by some candidate sets. Moreover, we guarantee that $T_{u_1} \cup T_{u_2}$ are connected to $T \setminus T_v$ if v is covered by some GREEN edge or candidate set. Therefore, all of T_{u_1}, T_{u_2} and $T \setminus T_v$ are interconnected after the iteration on v.

If the second case happens, i.e., v has more than 2 children. Since we omit all edges attached to v in this iteration, it is easy to see that BT_v is connected even though all of v_i's are removed. So we can conclude the correctness of CSFA-2VC. □

We have mentioned that CSFA-2VC can achieve 3.67-approximation for DPA-2VC. The approximation ratio is concluded as follows.

Theorem 2 *The CSFA-2VC computes a* 3.67*-ratio approximation of DPA-2VC problem.*

Proof (sketch) The idea is almost the same as the proof of Theorem 1. The key is to prove $|U_2| \leq 2opt_v$. It follows: 1. each candidate node set has at least one node appeared in an optimal solution; 2. two candidate node sets are disjoint. One can check all arguments for CSFA still hold for CSFA-2VC. We omit details here due to page limit. □

6 Conclusion

In this paper, we consider the dual power model and address the dual power assignment problem for 2-edge connectivity and 2-vertex connectivity. Due to the NP-hardness for both problems, we propose the Candidate Sets Filtering Algorithms (CSFA). We also prove that our algorithms can achieve 3.67-approximation for both problems, which improve 4-approximation algorithms in previous works.

For future research, we want to consider the geometry nature of the problem, for example Dual Power Assignment problems in Euclidean disk graphs. Up to our best known, there is no better result achieved for the 2-dimensional Euclidean version.

In addition, an algorithm for node-weighted version of DPA problem is also valuable. The weight of a node may be regarded as some priority according to the different functions, positions and remaining energy. It can also very useful to develop some scheduling mechanism in the dual power model.

References

Calinescu G, Wan P-J (2006) Range assignment for biconnectivity and k-edge connectivity in wireless ad hoc networks. Mob Netw Appl 11:121–128

Chen J-J, Lu H-I, Kuo T-W, Yang C-Y, Pang A-C (2005) Dual power assignment for network connectivity in wireless sensor networks. In: Proceedings of the IEEE GLOBECOM

Clementi AEF, Penna P, Silvestri R (1999) Hardness results for the power range assignment problem in packet radio networks. In: Proceedings of the 2nd international workshop on approximation algorithms for combinatorial optimization problems (APPROX), pp 197–208

Diestel R (2000) Graph theory, 2nd edn. Springer, Berlin

Hajiaghayi M, Immorlica N, Mirrokni VS (2003) Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In: Proceedings of the 9th annual international conference on mobile computing and networking

Kirousis LM, Kranakis E, Krizanc D, Pelc A (2000) Power consumption in packet radio networks. Theor Comput Sci 243:289–305

Lloyd EL, Liu R (2006) Approximating the minimum number of maximum power users in ad hoc networks. Mob Netw Appl 11:129–142

Lloyd E, Liu R, Marathe M, Ramanathan R, Ravi SS (2002) Algorithmic aspects of topology control problems for ad hoc networks. In: Proc. third ACM international symposium on mobile ad hoc networking and computing (mobiHoc), pp 123–134

Park M-A, Wang C, Willson III JKV, Wu W, Farago A (2006) Fault-tolerant dual-power assignment in wireless sensor networks, Technical Report, UTDCS-52-06

Poojary N, Krishnamurthy SV, Dao S (2001) Medium access control in a network of ad hoc mobile nodes with heterogenous power capabilities. In: IEEE ICC'01, pp. 872–877

Rong Y, Choi H, Choi H-A (2004) Dual power management for network connectivity in wireless sensor networks. In: Proceedings of the 18th international parallel and distributed processing symposium

Shah V, Krishnamurthy SV, Poojary N (2004) Improving the MAC layer performance in ad hoc networks of nodes with heterogeneous transmit power capabilities. In: IEEE ICC'04

Wang C, Park M-A, Willson III JKV, Wu W, Farago A (2008) On approximate optimal dual power assignment for biconnectivity and edge-biconnectivity. Theor Comput Sci 396:180–189