Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 396 (2008) 180–190

Theoretical Computer Science

www.elsevier.com/locate/tcs

On approximate optimal dual power assignment for biconnectivity and edge-biconnectivity☆

Chen Wang[a,*], Myung-Ah Park[b], James Willson[b], Yongxi Cheng[a], Andras Farago[b], Weili Wu[b]

[a] *Institute for Theoretical Computer Science, Tsinghua University, China*
[b] *University of Texas at Dallas, United States*

Received 6 November 2007; received in revised form 7 January 2008; accepted 17 January 2008

Communicated by D.-Z. Du

Abstract

Topology control is one of the major approaches to achieve energy efficiency as well as fault tolerance in wireless networks. In this paper, we study the *dual power assignment problem* for 2-edge connectivity and 2-vertex connectivity in the symmetric graphical model.

The problem has arisen from the following practical 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-tolerant 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. Note that to achieve a minimum number of high-power nodes is harder than an optimization problem in the same model whose objective is to minimize the total power cost.

We first address these two optimization problems (2-edge connectivity and 2-vertex connectivity version) under the general graph model. Due to the NP-hardness, we propose an approximation algorithm, called *prioritized edge selection algorithm*, which achieves a 4-ratio approximation for 2-edge connectivity. After that, we modify the algorithm to solve the problem for 2-vertex connectivity and also achieve the same approximation ratio. We also show that the 4-ratio is tight for our algorithms in both cases.

Keywords: Dual power assignment; 2-edge connectivity; 2-connectivity; Approximation algorithms

1. Introduction

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

☆ This work was partially supported by National Natural Science Foundation of China under grant No. 60553001 and National Basic Research Program of China under grant No. 2007CB807900, 2007CB807901.

* Corresponding author. Tel.: +86 1062797304.
E-mail address: wc00@mails.tsinghua.edu.cn (C. Wang).

doi:10.1016/j.tcs.2008.01.029

Since a mobile device is usually battery-powered and vulnerable, energy efficiency and fault tolerance are two main issues in a wireless ad hoc network. They are even more critical in a wireless sensor network which is regarded as an application of ad hoc networks.

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 the 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 asymmetry 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 symmetrize edges among neighbor nodes. Here, we assume that there exists a *(symmetric) link* between nodes u and v if $d(u, v) \leq r_u$ and $d(u, v) \leq r_v$ 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 have 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 in which there are only two universal transmission power levels. With this dual power assignment, a 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 the 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 [13] and [14] 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 minimizes the number of high power nodes and meanwhile produces 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 a reduction from the minimum set cover problem, which is well-known NP-hard. The proof is not very hard and therefore omitted.

The rest of the paper is organized as follows. We firstly introduce some related work in Section 2. Then, in Section 3, some preliminaries and basic definitions of our algorithm are given. Our approximation algorithms are formally described in Sections 4 and 5. Specifically, in Section 4, we proposed the Prioritized Edge Selection Algorithm for the DPA-2EC problem followed by a proof of the approximation results, and in Section 5, we arrive at the 4-approximation for DPA-2VC by using a modified PESA algorithm. As the last part, we conclude the paper and give some future works in Section 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 [11], some approximation algorithms are developed. Kirousis et al. [15] give a 2-approximation algorithm by constructing the minimum spanning tree of the network graph. Calinescu et al. [3] improve the approximation factor to 1.69 with a Steiner tree based algorithm. As a natural generalization of connectivity, the 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. [9] present a 8-approximation algorithm. For the general k-connectivity and k-edge connectivity problem, an $O(k)$-approximation centralized algorithm was designed by Hajiaghayi et al. [5]. Calinescu et al. [3] improve the approximation result for k-edge connectivity to 2k, and for biconnectivity to 4.

To the best of our knowledge, the dual power assignment problem was first studied by Rong et al. [4]. 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. [2] improve the approximation factor of asymmetric version to 1.75. In [1], Lloyd and Liu study the minimum number of maximum power users problem, which is actually the symmetric version of 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. To the best of knowledge, the only work where fault tolerance is taken in account is [12]. The authors proposed the minimum spanning tree augmentation algorithms for 2-edge connectivity and 2-vertex connectivity with dual power assignment. The algorithms achieve an approximation ratio of 6 for 2-edge connectivity and 5 for 2-vertex connectivity respectively.

In this paper, we study the dual power assignment problem both for 2-edge connectivity(DPA-2EC) and 2-vertex connectivity(DPA-2VC) in general undirected graphs. Our algorithm can achieve a better approximation ratio of 4, for both DPA-2EC and DPA-2VC.

3. Preliminaries

Before we introduce our algorithm, let us formally define some concepts and describe some basic properties without proof which are important for our algorithm. Although some of them can be found from text books of graph theory, e.g. [16], the definitions may be different.

Definition 2. An undirected loopless multi-graph[1] $G = (V, E)$ is called **2-edge connected** (**2-vertex connected**, resp.) if, G is a single node, or two nodes with at least two edges between them, or has at least 3 nodes and remains connected after removing any edge (vertex, resp.).[2]

Definition 3. A connected component of a graph is one of its maximal connected subgraphs. Similarly, a 2-edge connected (2-vertex connected, resp.) component is a maximal 2-edge connected (2-vertex connected, resp.) subgraph.

Definition 4. Given a graph G, a **bridge** is an edge that separates its two endpoints. A **cut vertex** is a vertex which separates two other vertices of the same connected component.

Definition 5. Let G be a graph. A **E-block** is a maximal connected subgraph without bridges. A **block** is a maximal connected subgraph without cut vertex and bridges.

It can be shown that each E-block of a graph is a 2-edge connected component, and each block is a 2-vertex connected component.

By maximality, different E-blocks are vertex-disjoint and different blocks overlap in at most one vertex which is a cut vertex of G. Actually, the 2-edge connectivity (2-vertex connectivity, resp.) of a graph can be described by all its E-blocks (blocks and cut vertices, resp.) together with its bridges (cut vertices, resp.).

[1] It means multiple edges between two nodes are allowed. Note that even though the input graphs are both simple graphs, we need use a multi-graph in the intermediate process.

[2] Note that in our definition, we also consider a single node as a 2-vertex and 2-edge connected graph for convenience.

Definition 6. Let G be a graph, A denotes the set of bridges of G and $\mathcal{B}$ denotes the set of its E-blocks. The **E-block graph** of G, denoted by $EBG(G)$, is the graph on $\mathcal{B}$ formed by the edges (B_1, B_2) if there exist some $v \in B_1$ and $u \in B_2$ and $(u, v) \in A$.

Definition 7. Let G be a graph, C denotes the set of cut vertices, A denotes the set of bridges of G and $\mathcal{B}$ denotes the set of its blocks. The **block graph** of G, denoted by $BG(G)$, is the bipartite graph on $C \cup \mathcal{B}$ formed by two types of edges:

1. (c, B) with $c \in B$ where c is any cut vertex and B is any block;
2. (B_1, B_2) if there exist some $v \in B_1$ and $u \in B_2$ and $(u, v) \in A$.

The concepts of E-blocks and blocks are illustrated in Figs. 1 and 2 respectively. The following proposition reduces the connectivity structure of a given graph to that of its E-blocks (blocks).

Proposition 1. *The E-block (block) graph of a graph is a forest. In addition, a graph is 2-edge (2-vertex) connected iff its E-block (block) graph is a single node.*

Given a graph G, we define a natural mapping $EB(\cdot)$ from its vertex set to the vertex set of $EBG(G)$ according to the E-block inclusion. Specifically, a vertex v is mapped to a E-block $EB(v)$ if $v \in EB(v)$. Similarly, for block graph, we can also define the mapping $B(\cdot)$ which maps a non-cut vertex to its corresponding block and is an identity mapping for cut vertices. Therefore, when we add an edge (u, v) into G, we can consider it as adding the edge $(EB(u), EB(v))$ into $EBG(G)$ and $(B(u), B(v))$ into $BG(G)$. The following proposition shows that they are equivalent.

Proposition 2. *Let G be an graph, u and v are two vertices of G. Then*

$$EBG(G \cup \{(u, v)\}) = EBG(EBG(G) \cup \{(EB(u), EB(v))\})$$

and

$$BG(G \cup \{(u, v)\}) = BG(BG(G) \cup \{(B(u), B(v))\}).$$

4. Approximation algorithm for DPA-2EC

In this section, we propose an algorithm called the *Prioritized Edge Selection Algorithm* (PESA) to solve the DPA-2EC problem.

The basic idea of the algorithm is as follows. In order to achieve 2-edge connectivity, we begin with G_l and keep adding edges from $E_h \setminus E_l$ into it. For the resultant graph, denoted by G', we can claim that it is already 2-edge connected if $EBG(G')$ is a single node according to Proposition 1.

We introduce the following terminology to describe the algorithm.[3]

- A **covered edge** is an edge in G' with its two endpoints corresponding to the same E-block node.
- A **candidate edge** is an edge which is in $G_h \setminus G'$ and adding which into G' makes some uncovered edges covered.
- A **leaf** is a E-block of degree less than 2 in $EBG(G')$.
- An **isolated leaf** is a leaf without edges to any other E-blocks.
- A **non-trivial tree** is a connected component of $EBG(G')$ with at least 2 E-blocks.
- A **twig** with regard to a leaf C is a maximal subtree of the non-trivial tree containing C, where each node except C has degree 2.
- A **back edge** is a candidate edge connecting a leaf to a different node in the same twig.
- A **line path** is a non-trivial tree where there are exactly two leaves.

In order to minimize the number of selected high power nodes, we need to choose edges in an effective way. To do this, we assign an priority to each candidate edge which is not yet in G'.

Definition 8 (*Edge Prioritization, see Fig. 3*).

- A **Type-1 edge** is an edge such that, if added, the number of leaves in $EBG(G')$ decreases. We assign high priority to the Type-1 edges.

[3] Note that we will adopt all this terminology with respect to BG(G) in the next section.

- A **Type-2 edge** is a back edge of $EBG(G')$ such that no other back edge incident to the same leaf covers more edges. A Type-2 edge is non-leaf-reducing edge by the definition of a twig.[4] We assign medium priority to the Type-2 edges.
- A **Type-3 edge** is an edge that connects an isolated leaf to an isolated leaf. The Type-3 edge set is assigned low priority.

Now, let us describe how PESA works in detail. In each iteration, we firstly choose an edge arbitrarily from the non-empty edge set with highest priority. Then we set two end-points of this edge to high power level and add this edge into G'. Note that we need to consider all edges which are induced by existing high power nodes. Lastly we update all edge priorities with respect to new G'. The algorithm terminates once $EBG(G')$ becomes a single node.

Algorithm 1. Prioritized Edge Selection Algorithm

```
INPUT: (V, E_h, E_l)
OUTPUT: Power assignment A such that the symmetric graph induced by A is 2-edge connected.
A(v) ← r_l for all v ∈ V and initialize G' ← (V, E_l);
E ← E_h − E_l
Let Q_i be the set of Type-i edges where i = 1, 2, 3;
while EBG(G') has more than 1 node do
    Construct Q_i from E according to EBG(G') where i = 1, 2, 3;
    if Q_1 is not empty then
        choose arbitrary e = (u, v) from Q_1;
    else if Q_2 is not empty then
        choose arbitrary e = (u, v) from Q_2;
    else
        choose arbitrary e = (u, v) from Q_3;
    end if
    A(u) ← r_h;
    A(v) ← r_h;
    E ← E − {e};
    Update G' according to A;
end while
return A;
```

Updating G' and Q_i (the set of Type-i edges) in each iteration can be done by a depth first search which needs $O(n^2)$ time and the algorithm always terminates in n iterations because we add at least one more high power node in each iteration. Thus, the time complexity of PESA is at most $O(n^3)$.

The following theorem shows that PESA can achieve a good approximation compared to the optimal solution, which is one of the main results of this paper.

Theorem 1. *PESA computes a 4-approximation for DPA-2EC.*

Note that in each iteration of the algorithm, although we select only one edge, more than 1 edge may be added and make the contraction more effective. However, this makes the analysis unnecessarily complicated. Therefore we only consider and count the selected edge instead of taking all the edges induced by the two high-power nodes into account. So G' in the following context is actually updated by selected edges rather than selected nodes. This may cause a problem in that for some iterations we select an edge without adding high power nodes. We can just omit such an iteration because it doesn't affect the output vertex set.

Before we start to prove this theorem, we need to define some notation.

- U: the set of high-power nodes in an optimal solution
- N_{leaf} : the number of leaves in $EBG(G')$
- N_1: the number of Type-1 edges selected
- N_2: the number of Type-2 edges selected
- S_3: the first iteration where a Type-3 edge is selected

[4] It is remarkable that adding a Type-2 edge does not reduce the number of leaves because it only reduces the length of the twig.

- T: the number of connected components in G' at S_3
- M_1: the number of Type-1 edges which connect two connected components of G' and is selected after S_3
- M_2: the number of Type-2 edges selected after S_3
- N_3: the number of Type-3 edges selected after S_3

Since once we select a candidate edge, we assign its two endpoints to the high-power level, the number of total high-power nodes is at most $2(N1 + N2 + N3)$. In order to prove Theorem 1, we will give a bound for each of N_i's by proving Lemmas 1–9.

Lemma 1. *Given a 2-edge connected graph $G = (V, E)$ and any non-trivial vertex partition $(U, V - U)$ (neither U nor $V - U$ is empty), there are at least two edges between U and $V - U$.*

Proof. Assume that for some vertex subset $U \subseteq V$, there are less than two edges between U and $V - U$. It means that either there are no edges between U and $V - U$, which implies G is not connected, or only one such edge exists whose removal will disconnect U and $U - V$, which implies that G is not 2-edge connected.

Therefore, there must be at least two edges between U and $V - U$, and this holds for every $U \subseteq V$. □

Lemma 2. $|U| \geq N_1$.

Proof. At the beginning of the algorithm, all nodes are assigned to the low power level. By Lemma 1, each leaf in $EBG(G')$ must have at least one high-power node. In other words, $|U| \geq N_{leaf}$. Note that PESA considers only three types of edges at each iteration. When a Type-1 edge is selected, the number of leaves decreases by at least 1. When a Type-2 or Type-3 edge is selected, the number of leaves remains the same. This implies that $|N_{leaf}| \geq |N_1|$. Therefore the lemma holds. □

Lemma 3. *If a Type-3 edge is selected by PESA in any iteration, $EBG(G')$ must consist only of isolated E-blocks at the beginning of this iteration.*

Proof. According to the edge priorities, we will never select a Type-3 edge if either a Type-1 or a Type-2 edge exists. Suppose that there is a non-trivial tree in $EBG(G')$ and C is a leaf of the non-trivial tree. Since G_h is 2-edge connected, there must be a candidate edge $e \in E_h - E_l$ connecting C to another E-block C' in $EBG(G')$. If C' is in a different tree, e must be a Type-1 edge. On the other hand, if C' is within the same tree, e must be either a Type-1 edge or a Type-2 edge. Thus PESA should not select a Type-3 edge at this iteration, which leads to a contradiction. □

Lemma 4 (*Structure Property of $EBG(G')$*). *At any iteration after S_3, $EBG(G')$ consists of isolated leaves and up to one line path.*

Proof. We prove the structure property by induction. By Lemma 3, the structure property holds at S_3. Suppose that the structure property holds at the beginning of i-th iteration after S_3. Then $EBG(G')$ consists of (a) either isolated nodes or (b) only one line path and isolated nodes. First, assume that (a) is the case. Then the Type-1 and Type-2 edge pools must be empty at the i-th iteration, and PESA selects a Type-3 edge to connect two isolated E-blocks. This results in a line path with length one and remaining isolated E-blocks in $EBG(G')$ at the beginning of $(i + 1)$-th iteration. Therefore, the lemma holds for the case (a).

Assume that (b) is the case and P is the line path. We know that a leaf C of the path P must have a candidate edge e which connects C to another E-block C' in $EBG(G')$. C' can be either on the path or an isolated leaf.

For both cases, we will get either isolated leaves or a line path and isolated leaves at the beginning of $(i + 1)$-th step. Therefore, the lemma holds. □

Lemma 5. *If there exists a path with length of l at the beginning of any iteration after S_3, the path must consist of at least $l - 1$ Type-1 edges and at most one Type-3 edge.*

Proof. Note that a Type-2 edge is immediately covered once it is added into G'. So a line path is composed by Type-1 and Type-3 edges. However, a Type-3 edge is only added if there exist only isolated leaves in $EBG(G')$ and thus there cannot be two uncovered Type-3 edges in $EBG(G')$. This implies the lemma immediately. □

Lemma 6. $T \geq M_1 + N_3$.

Proof. Every Type-3 edge reduces the number of trees by 1 and every Type-1 edge connecting two trees reduces the number of trees by 1. Type-2 edge and intra-tree Type-1 edge does not reduce the number of trees. So, $M_1 + N_3 = T - 1$. □

Lemma 7. *The edge selected at the last iteration must be a Type-1 edge which is an edge connecting two endpoints of a line path.*

Proof. An observation that neither Type-2 edge nor Type-3 edge can contract $EBG(G')$ into a single node implies the lemma immediately. □

Lemma 8. $M_1 \geq M_2$.

Proof. By Lemma 5, any line path after $S3$ consists of Type-1 edges and up to one Type-3 edge. In particular, a line path can be only expanded by Type-1 edges, but can be contracted by Type-2 edges and Type-1 edges. Because a path is eliminated into an isolated leaf by a Type-1 edge according to a similar argument to Lemma 7, and also a Type-2 edge covers at least one edge in the path, We can see that the number of Type-1 edges which expand a line path is no less than the number of Type-2 edges which contract the line path. Summing over all appeared line paths after $S3$, we have $M_1 \geq M_2$. □

According to Lemmas 6 and 8, we can immediately get the following corollary.

Corollary 1. $T \geq M_2 + N_3$.

Lemma 9. $|U| \geq T + N_2 - M_2$.

Proof. Note that $N_2 - M_2$ represents the number of Type-2 edges selected before S_3 by the definitions of N_2 and M_2.

Suppose PESA selects a Type-2 edge e incident to a leaf C before S_3. C must not have any Type-1 edge, since a Type-1 edge overrides a Type-2 edge. This implies that any node in C has no incident edge connected to any other tree, nor does it have an edge to a E-block outside its twig. However, C must have at least two adjacent edges by Lemma 1. Thus, it is necessary for U to contain at least one high-power node v in $C' \subset C$, which consists of nodes with at least one candidate edge adjacent to a node outside C. After the Type-2 edge is added, as no other edge incident to C covers more edges, no nodes in C' have an candidate edge. Thus all C's during the algorithm are disjoint. Let W be the intersection of U and the union of all C'. Then $|W| \geq N_2 - M_2$.

Let us consider an inter-tree edge e at iteration i. Then, at iteration i' where $i' < i$, it is also an inter-tree edge, since a tree never shrinks over iterations. In other words, if an edge e is not an inter-tree edge at iteration i', it can not be an inter-tree edge at iteration executed later than i'. Since a node v in W has no inter-tree adjacent edge at some iteration before S_3 as mentioned above, any edge adjacent to v can not be an inter-tree edge at any iteration after S_3.

Now, consider iteration S_3. By Lemma 4 and the definition of T, $EBG(G')$ consists of T isolated leaves, each of which forms a tree. By Lemma 1, we know that U must have at least one high-power node in each isolated leaf in order to make the isolated leaves 2-edge connected. However, any v in W can not be used to connect any two isolated leaves. Otherwise, it implies that v has an inter-tree edge incident to it, contradicting the definition of W. Therefore, $|U| \geq T + |W| \geq T + N_2 - M_2$. □

Proof of Theorem 3. Combining Lemmas 2 and 9 and Corollary 1, we know that

$$N_1 + N_2 + N_3 \leq N_1 + (M_2 + M_3) + N_2 - M_2 \leq N_1 + T + N_2 - M_2 \leq 2|U|.$$

Since each selected edge may increase the number of output high power nodes by at most 2, we therefore conclude the 4-approximation ratio of PESA immediately. □

Actually, the analysis of 4-approximation ratio of PESA is tight. Fig. 4 shows a network of size $5n$ which provides the approximation ratio of 4. Each solid circle represents a clique when low power is assigned to every node in the network. There is no low-power edge between any two solid circles. The dashed circle represents a clique when high power is assigned to every node in the network. In this instance, the optimal solution of the DPA-2EC problem is to assign high power to one node in each c_i where $i = 1, \ldots n$. However, PESA obtains $4(n - 1)$ high-power nodes in the worst case.

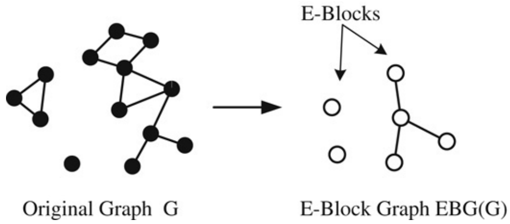

Fig. 1. E-block and E-block graph.

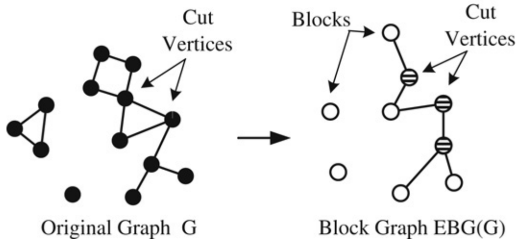

Fig. 2. Block and block graph.

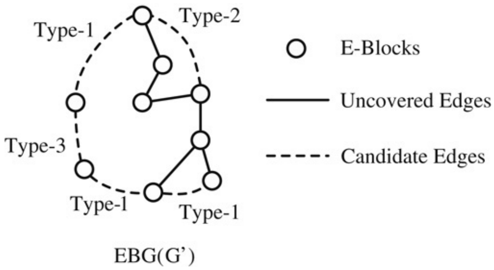

Fig. 3. Edge priorities of PESA.

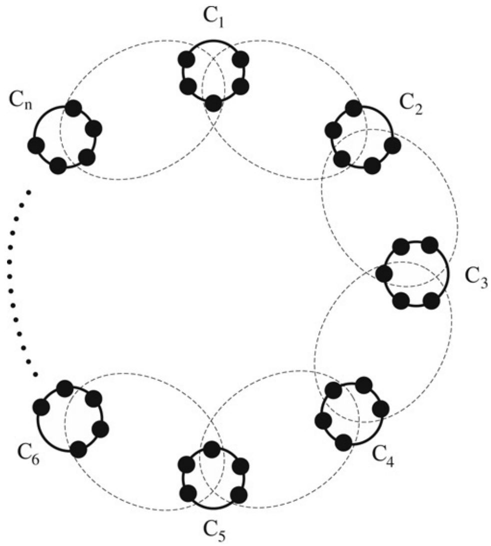

Fig. 4. An instance with the approximation ratio of 4.

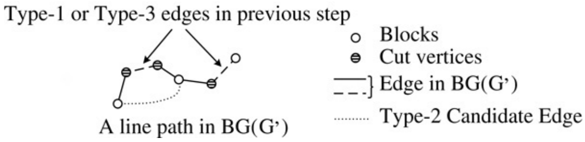

Fig. 5. An Type-2 Edge after S_3 always covers either a Type-1 edge or a Type-3 edge.

5. Modified PESA algorithm for DPA-2VC

In the section, we modify PESA to solve the DPA-2VC problem. The basic idea is quite similar to what we did in the previous section.

The edge priorities and the algorithm are defined almost the same as last section except replacing concepts for 2-edge connectivity by corresponding concepts for 2-vertex connectivity. We need emphasize that a block graph consists of blocks and cut vertices whereas a E-block graph consists of E-blocks only, and therefore we need replace E-blocks with blocks and cut vertices together. The following theorem gives a similar result just as Theorem 1.

Theorem 2. *Modified PESA computes a 4-approximation of DPA-2VC in $O(n^3)$ time.*

Proof Sketch. First of all, we define U, N_i $(i = 1, 2, 3)$, M_j $(j = 1, 2)$, and the iteration S_3 similarly. We need to prove that

$$N_1 + N_2 + N_3 \leq 2|U|.$$

We will adopt the Lemmas 2, 6, 8 and 9. Consequently, the 4-approximation ratio is derived similarly.

The time complexity is also $O(n^3)$ due to the $O(|E|)$ time depth-first-search algorithm for finding all the 2-vertex connected components and cut vertices. □

In order to accomplish the proof, we need to reprove all the Lemmas we have proved in Section 4 under the 2-vertex connectivity context. In fact, almost all proofs can be immediately modified by some simple replacements of terms, except the following new Lemma 8.

Lemma 8 (’). $M_1 \geq M_2$.

Proof. In the 2-vertex connectivity case, a line path is not only formed by Type-1 edges and at most one Type-3 edge, but also by edges between cut vertices and blocks. However, note that a Type-2 edge will always cover a Type-1 edge or a Type-3 edge. Otherwise, G' is invariant after the edge added. This is illustrated in Fig. 5.

Moreover, the eliminating operations from a line path to an isolated block must be achieved by a Type-1 edge. Therefore, $M_1 \geq M_2$. □

Similarly, the approximation ratio is also tight. A counter example is shown in Fig. 6. The network consists of $12n$ vertices. The low-power graph has $14n$ edges which are represented by solid lines. High power edges are drawn as dotted lines. In addition, the graph has n connected components each of which consists of three 2-connected components. All the high power edges are labeled into four classes: $\{e_i | i = 1 \ldots n\}$, $\{a_{ij} | i = 1 \ldots n, j = i + 1\}$, $\{b_{ij} | i = 1 \ldots n, j = i + 1\}$ and $\{c_{ij} | i = 1 \ldots n, j = i + 1\}$.[5] The optimal solution is to assign high-power levels to each node attached by some c_{ij}. Therefore, the optimal solution has $2n$ nodes.

Consider the modified PESA algorithm. At the beginning, each e_i is a Type-1 edge and can be selected. After that, G' has n isolated nodes. From then, according to the algorithm, one can check that a possible order of edge selection is $a_{12}, b_{12}, a_{23}, b_{23}, \ldots, a_{i(i+1)}, b_{i(i+1)}, \ldots, a_{n1}, b_{n1}$. Therefore, we eventually add $8n$ nodes in total, which is exactly 4 times the optimal solution.

6. Conclusion

In order to reduce the power consumption and improve the fault tolerance in wireless network, we consider the dual power model and address the dual power assignment problem for 2-vertex connectivity and 2-edge connectivity.

[5] Note that $a_{n(n+1)} = a_{n1}$, $b_{n(n+1)} = b_{n1}$ and $c_{n(n+1)} = c_{n1}$.

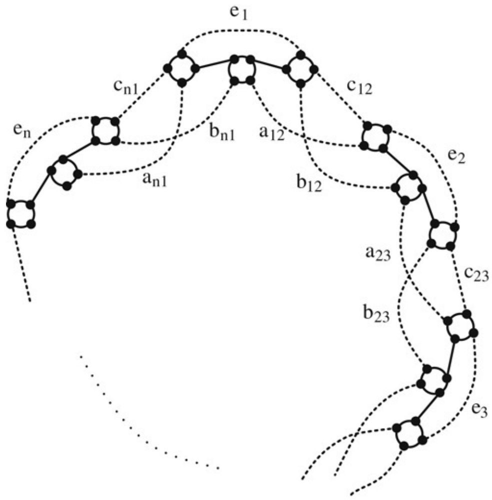

Fig. 6. An instance with the approximation ratio of 4 for modified-PESA.

Due to the NP-hardness for both problems, we propose the Prioritized Edge Selection Algorithm to achieve 4-approximation for both problems. We also show that this ratio is tight with respect to PESA.

In future work, we plan to develop some distributed version of our algorithm. Another interesting problem is to consider the geometric nature of the problem, for example Dual Power Assignment problems in a unit disk graph. Up to our best knowledge, there is no better result achieved for this special version. It would be also interesting to investigate the problem when more than two power levels are introduced.

References

[1] E.L. Lloyd, R. Liu, Approximating the minimum number of maximum power users in ad hoc networks, Mobile Networks and Applications 11 (2006) 129–142.

[2] Jian-Jia Chen, Hsueh-I Lu, Tei-Wei Kuo, Chuan-Yue Yang, Ai-Chun Pang, Dual power assignment for network connectivity in wireless sensor networks, in: 2005 Proceedings of the IEEE GLOBECOM, 2005.

[3] Gruia Calinescu, Peng-Jun Wan, Range assignment for biconnectivity and k-edge connectivity in wireless ad hoc networks, Mobile Networks and Applications 11 (2006) 121–128.

[4] Yanxia Rong, Hongsik Choi, Hyeong-Ah Choi, Dual power management for network connectivity inwireless sensor networks, in: Proceedings of the 18th International Parallel and Distributed Processing Symposium, 2004.

[5] M. Hajiaghayi, Nicole Immorlica, Vahab S. Mirrokni, 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, 2003.

[6] P. Sholander, G. Frank, A. Yankopolus, Energy-efficient network techniques for wireless sensor networks, in: Proceedings of the IEEE Milicom Conference, 2003.

[7] R. Ramanathan, R. Rosales-Hain, Topology control of multihop wireless networks using transmit power adjustment, in: INFOCOM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Socienties, Proceedings, in: IEEE, vol. 2, March 2000, p. 403C413.

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

[9] E. Lloyd, R. Liu, M. Marathe, R. Ramanathan, S.S. Ravi, Algorithmic aspects of topology control problems for ad hoc networks, in: Proc. Third ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc, 2002, pp. 123–C134.

[10] Xiaohua Jia, Dongsoo Kim, Sam Makki, Peng-Jun Wan, Chih-Wei Yi, Power assignment for k-connectivity in wireless ad hoc networks, Journal of Combinatorial Optimization 9 (10) (2005) 213–222.

[11] A.E.F. Clementi, P. Penna, R. Silvestri, 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, 1999, pp. 197–208.

[12] M.A. Park, C. Wang, J.K.V. Willson III, W. Wu, A. Farago, Fault-tolerent dual-power assignment in wireless sensor networks, Technical Report, UTDCS-52-06, 2006.

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

[14] V. Shah, S.V. Krishnamurthy, N. Poojary, Improving the MAC layer performance in Ad Hoc networks of nodes with heterogeneous transmit power capabilities, in: IEEE ICC'04, pp. 2004.

[15] L.M. Kirousis, E. Kranakis, D. Krizanc, A. Pelc, Power consumption in packet radio networks, Theoretical Computer Science 243 (2000) 289C305.

[16] Reinhard Diestel, Graph Theory, Springer.