A Simpler Constant Factor Approximation for The k-connected m-domination Set Problem in Unit Disk Graph

Bei Liu*, Wei Wang*, Donghyun Kim†, Yingshu Li‡, Sung-Sik Kwon§
* School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China.
E-mail: liubei@stu.xjtu.edu.cn, wang_weiw@163.com
†Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA.
Email: donghyun.kim@kennesaw.edu
‡ Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA.
E-mail: yli@cs.gsu.edu
§ Department of Mathematics and Physics, North Carolina Central University, Durham, NC 27707, USA.
E-mail: skwon@nccu.edu

***Abstract*—Over years, many efforts are made for the problem of constructing quality fault-tolerant virtual backbones in wireless network. In case that a wireless network consists of physically equivalent nodes, e.g. with the same communication range, unit disk graph (UDG) is widely used to abstract the wireless network and the problem is formulated as the minimum k-connected m-dominating set problem on the UDG. So far, most results are focused on designing a constant factor approximation algorithm for this NP-hard problem under two positive integers k and m satisfying $m \geq k \geq 1$ and $k \leq 3$. Very recently, Shi et. al. and Fukunaga separately introduced constant factor approximation algorithms for the problem with $m \geq k \geq 1$. However, we found the structures of the algorithms are extremely complicated, and thus it would be difficult to implement and use them in practice. Motivated by such observation, this paper introduces a novel approximation algorithm for the problem with $m \geq k \geq 1$. This algorithm is based on our new technique which first computes a 1-connected m-dominating set D and repeatedly (a) decomposes D into an i-connected block tree, with $i = 2, 3, \cdots, k$, and (b) use this graph structure to improve the connectivity of D, until D becomes k-connected. We provide a rigorous theoretical analysis to prove that the proposed algorithm is correct and its approximation ratio is a constant. We compare the structure of our algorithm against the existing ones and show our algorithm is much simpler to understand and implement.**

***Index Terms*—Wireless Sensor networks, approximation algorithm design, graph theory, connected dominating set, virtual backbone.**

I. Introduction

Wireless networks such as wireless ad-hoc networks and wireless sensor networks are composed of numerous wireless mobile nodes, and have a number of important applications such as environment and habitat monitoring, traffic control, health applications, etc. [13], [14]. In most cases, a wireless node is battery operated and thus has a limited power source. In wireless communication, the amount of energy consumed for a node to transmit a message to another node increases super-linearly proportional to the distance between them. As a result, most wireless networks prefer multi-hop communication over long range direct communication to conserve its energy.

Unfortunately, the multi-hop communication strategy increases the number of messages flying over the network drastically and causes a huge amount of wireless signal interference and collision. As a result, the nodes consume much of its energy for retransmitting messages and waste lots of energy. This problem is known as the broadcasting storm problem and is a serious but inheriting issue in most multi-hop routing wireless networks [15]. To ease the impact of the problem, one promising strategy would be having a backbone-like structure in the wired counterpart so that the number of nodes which are involved in the routing can be reduced. Based on the observation, Ephremides et. al [16] suggested to establish a subset of wireless nodes to be in charge of routing messages while the other nodes are not. Nowadays, this subset is called as a virtual backbone (VB) of the wireless network. Recent studies show that in addition to improve the efficiency of the wireless network, virtual backbone is known to bring several advantages to wireless networks as its adoption can be used to alleviate routing overhead and serve as an efficient platform for unicast, multicast, and fault-tolerant routing.

A subset of nodes in the unit disk graph (UDG) representing a wireless network of interest can be a VB in the graph if (a) the subgraph of the UDG induced by the subset is connected and (b) all nodes are either in the subset or adjacent to a node in the VB. In theory, the subset of a graph satisfying the requirements is referred as the connected dominating set (CDS). Apparently, a CDS of a UDG is better than another CDS of the UDG if its size is smaller as that means the CDS will suffer less from wireless signal interference and collision. Thus, Guha and Khuller [17] modelled the problem of computing quality virtual backbone as the minimum connected dominating set (MCDS) problem. The (minimum) dominating set problem is a well-known NP-hard problem and a generalized version of the MCDS problem as it does not require the induced graph by the subset connected. As a result, the MCDS problem is also NP-hard, which implies that it is impractical to compute an optimal solution of a given MCDS problem, i.g. a minimum size CDS. As a result, many efforts are made to design and analyze an approximation algorithm for the problem, which has a worst-case performance guarantee [1], [2], [3], [4], [5].

In many wireless networks such as wireless sensor networks, nodes are subject to fail due to many reasons such as battery exhaustion or hostile environment. Due to the reason, the virtual backbone for wireless network is desirable to have some degree of fault-tolerance, in particular against node failure. In theory, a graph is k-vertex-connected if the graph is still connected after the removal of any $k-1$ nodes, and a CDS whose induced graph has a higher level of fault-tolerance against such node failure. In [22], Dai and Wu investigated the requirements for a fault-tolerant virtual backbone and introduced the concept of the k-connected k-dominating set, or in short, (k,k)-CDS for a given k for the first time in the literature. In the later discussions, this concept is generalized as the k-connected m-dominating set, (k,m)-CDS. Formally speaking, a node subset is a k-connected m-dominating set if the following requirements are satisfied: given a network graph $G=(V,E)$ such as UDG, where V is the set of nodes and E is the set of edges, (a) $G[D]$, the subgraph of G induced by a node set D, is k (vertex) connected, i.e. $G[D \setminus D']$ is still connected for any $D' \subseteq D$ such that $|D'| \leq k$, and (b) D is a m-dominating set in G, i.e. for any $u \in V \setminus D$, u has at least m neighbors in D.

The minimum (cardinality) (k,m)-CDS problem is NP-hard as its special case with $k=1, m=1$, which is the MCDS problem, is NP-hard. Over recent years, many efforts are made to design a constant factor approximation algorithm to construct highly fault-tolerant virtual backbone. Note that by definition, $m \geq k \geq 1$ is desirable, otherwise, the failures of m nodes will fail the virtual backbone (i.e. some nodes will lose all of its neighboring backbone nodes) even though the backbone is fully operational (i.e. the subgraph induced by the residual backbone nodes is still connected). Wang et al. [36] proposed a 2-approximation algorithm for computing the minimum $(2,1)$-CDS problem; Shang et al. [34] introduced a constant factor approximation algorithm for the minimum $(2,m)$-CDS problem, where $m \geq 1$. At INFOCOM 2010, 2015, 2016, Kim et al. [12],Wang et al. [9], Zhang et.al [10] studied the problem of constructing a constant factor approximation algorithm for the minimum $(3,m)$-CDS problem, where $m \geq 3$. However, the question of designing a constant factor approximation algorithm for the minimum (k,m)-CDS problem is challenging for any $m \geq k \geq 4$.

To the best of our knowledge, there are three constant factor approximation algorithms to compute the minimum (k,m)-CDS, one of which by Shi et. al. [6] and two of which by Fukunaga [7]. While these are great results, the structures of the algorithms are extremely complicated, and thus it would be difficult to implement and use them in practice. To address this issue, this paper proposes a new approximation algorithm for the minimum (k,m)-CDS problem with $m \geq k \geq 1$. Roughly speaking, the algorithm is a round based one and each round consists of the following three steps, where i is initially 2 and grows up to k.

(a) compute $C_{i-1,m}$, which is a $(i-1,m)$-CDS of a given UDG,
(b) decompose the $C_{i-1,m}$ into components in which there exists no separating set of $C_{i-1,m}$ and add a bounded number of H-paths to connect these components, and
(c) add H-paths to connect components separated by the rest of separating sets.

As a result, we obtain a new simpler constant factor approximation algorithm for the minimum (k,m)-CDS problem given any m,k pair such that $m \geq k \geq 1$.

The reminder of this paper is organized as follows. Section II introduces some related work. Several important notations and definitions are provided in Section III. We present a new constant factor approximation algorithms for the minimum (k,m)-CDS problem in UDG in Section IV. We also give a theoretical analysis of its performance ratio. Section V is devoted to discuss the structural difference between the proposed algorithm and the existing alternatives. Finally, we conclude this paper and discuss some future research directions in Section VI.

II. Related work

Over many years, wireless networks have attracted lots of attention due to their useful applications [43], [44], [45]. During the past years, a lot of effort has been made to design approximation algorithms for fault-tolerant connected dominating set problems in wireless networks. Several approximation algorithms for constructing (k,m)-CDS have been proposed in the literature. The problem of constructing fault-tolerant virtual backbone was first proposed by Dai and Wu [22]. They proposed three heuristic algorithms o compute k-connected k- dominating set: a probabilistic algorithm k-Gossip; a deterministic algorithm, another probabilistic algorithm Color-Based k-CDS Construction. However, none of them guarantees the size bound of the resulting CDSs.

In [24], Alzoubi et al. proposed a approximation algorithm to construct a minimum CDS with performance ratio of 8. In [27], Li et al. provided a distributed algorithm for computing r-CDS whose performance ratio is 172. In [28], a localized algorithm was proposed and its performance ratio is 147. Wang et. al. introduced Connecting Dominating Set Augmentation (CDSA) in [36]. They proposed an 64-approximation centralized algorithm to construct a 2-connected 1-dominating virtual backbone. This algorithm first constructs a CDS, and then computes all the blocks and adds intermediate nodes to make backbone 2-connected. Further, in [34], Shang et al. proposed a centralized algorithm to construct 2-connected k-dominating set. They first choose an MIS of G, then choose an MIS k times, a k-dominating set D can be obtained, finally add H-path to make D 2-connected. This algorithm has performance ratio depending on k. Recently, Shi et al. proposed a greedy algorithm for computing minimum 2-connected m-dominating set [35], which has a performance ratio of 12.89 for $m \geq 5$.

Thai et. al. first introduced a centralized approximation algorithm to compute k-connected m-dominating set [33]. The main idea is: first compute a 1-connected m-dominating set; then calculate a k-connected k-dominating set based on the first step; at last, construct a k-connected m-dominating set. In [39], Wu et. al. proposed a centralized approximation algorithm, CGA, and a distributed algorithm, DDA, to construct k-connected m-dominating sets for any k and m. These two algorithms CGA, DDA are improved by the algorithms CGA and DDA, respectively in [40].

Recently, in [12], [41], the authors proposed a new polynomial time algorithm for computing $(3, m)$-CDS. They first, compute a $(2, 3)$-CDS, then iteratively convert the bad point to a good point by adding H-paths. The performance ratio of this algorithm is 280. This result is improved greatly by work [9], the first algorithm apply Tutte decomposition to design a basic algorithm, which first decomposing the $(2, m)$-CDS into bricks and add H-paths to construct the $(3, m)$-CDS; The second one is simpler to implement. The performance ratios of these algorithms are 87 and 62, respectively. In [10], Zhang et.al used the greedy strategy to design an algorithm for computing $(3, m)$-CDS, whose performance ratio is 26.34 for $(3, 3)$-CDS on UDG.

In graph theory, a block is a maximal component (i.e. connected subgraph) without separating set (a subset of nodes without which, the residual graph becomes disconnected). A block-tree of a graph is induced by the graph by contradicting every block into a node, and is a very useful tool to improve the connectivity of a (k, m)-CDS. For example, in [36], cut-vertices decompose a CDS into leaf-blocks and blocks to augment a $(1, 1)$-CDS to a $(2, 1)$-CDS. In [9], the unique Tutte decomposition of 2-connected graphs used to augment $(2, m)$-CDS to $(3, m)$-CDS. In this paper, we highly generalize this approach to obtain a constant factor approximation for the minimum k-connected m-dominating set problem in UDG based on the decomposition of k-connected graph throughout multiple round-based gradual augmentation of connectivity of a $(1, m)$-CDS to (k, m)-CDS.

Very recently, Shi et. al. [6] and Fukunaga [7] independently introduced three constant factor approximation algorithms for the problem with $m \geq k \geq 1$. Two of the algorithms (one by Shi et. al. and another by Fukunaga) aim to compute (k, m)-CDS directly instead of gradually augmenting connectivity of a $(1, m)$-CDS and are based on the algorithm to construct a subset k-connected subgraph by Nutov [8], whose structure is highly complicated. Another algorithm by Fukunaga uses a multiple round-based connectivity augmentation strategy like ours, but each round involves extremely higher computation than ours. We further discuss these differences in more detail in Section V.

III. Notations and Definitions

This section introduces several definitions and notations.

Definition 1 (Unit Disk Graph (UDG))**.** *A graph $G = (V, E) = (V(G), E(G))$ on a 2-dimensional euclidean space is referred as a UDG, if for any pair $u, v \in V$, there exists a bidirectional edge between them only if the euclidean distance between them is not greater than 1, i.e. $udist(u, v) \leq 1$.*

Definition 2 (k-Connected Graph)**.** *In graph theory, a graph $G = (V, E)$ is called a k-connected graph,if k is the size of the smallest subset of vertices such that the graph becomes disconnected if you delete them. An equivalent definition is that a graph G with at least two vertices is k-connected if, for every pair of its vertices, it is possible to find k node-disjoint paths connecting these nodes.*

Definition 3 (m-Dominating Set)**.** *Given a graph $G = (V, E)$, we call a vertex v dominated by a vertex u if there exists an edge $(u, v) \in E$. A subset $D \subset V$ is an m-dominating set if every vertex v in $V \setminus D$ is dominated by at least m vertices in D.*

Definition 4 (k-Connected m-Dominating Set)**.** *A subset $D \subset V$ is a k-connected m-dominating set ((k, m)-CDS) of graph $G = (V, E)$ if, (i) D is an m-dominating set of G and (ii) the induced subgraph $G[D]$ is k-connected.*

In this paper, (k, m)-CDS is abbreviated as $C_{k,m}$, and both represent a k-connected m-dominating set.

Definition 5 (Vertex Cut or Separating Set)**.** *$G = (V, E)$ is a k-connected graph, if there exists a subset $S = \{u_1, u_2, ..., u_k\} \subset V$ whose removal renders the induced graph $G \setminus \{u_1, u_2, ..., u_k\}$ disconnected, then S is called a vertex cut (or vertex separator, separating set).*

Definition 6 ((a, b)-Separator)**.** *In graph theory, given $G = (V, E)$, a vertex subset $S \subset V$ is a vertex cut (or vertex separator, separating set) for non-adjacent vertices a and b if the removal of S from the graph separates a and b into distinct connected components. Then we call S a (a, b)-separator.*

Definition 7 (H-path)**.** *Given a graph G, an H-path P of a subgraph H is a path between two different nodes in H such that no inner node of P is in H.*

In later discussion, given a graph $G = (V, E)$, we call a vertex v adjacent to a set of vertices $S = \{s_1, ..., s_k\}$ if there exists at least one node $s_i \in S$ such that edge $(v, s_i) \in E$. Further, we call a node set $V_1 = \{v_1, v_2, ..., v_s\}$ adjacent to another set of vertices $U_1 = \{u_1, ..., u_t\}$ if there exists at least an edge $(v_i, u_j) \in E$, where $v_i \in V_1$ and $u_j \in U_1$.

Definition 8 (Augmentation Problem)**.** *Given a k-connected graph $G = (V, E)$, and a $(k-1, m)$-CDS D of G, find a subset $H \subset V \setminus D$ with minimum size such that the induced subgraph $D \cup H$ is k-connected.*

Now, we introduce a decomposition of the k-connected graph, which plays an important role in the designing of our algorithm.

As we know, if a graph G is a k-connected graph, but not a $(k+1)$-connected graph, there must exist at least one

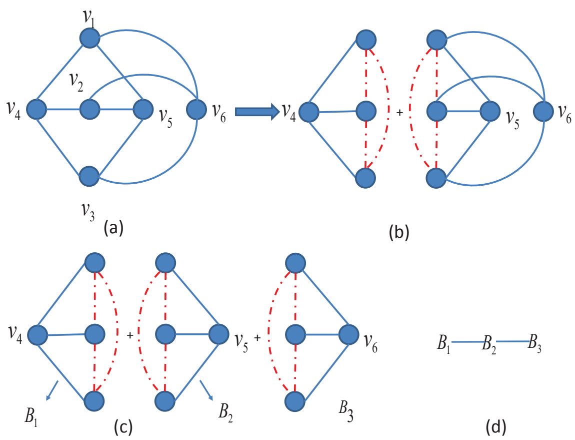


Fig. 1. (a) 3-connected graph, (b) $\{v_1, v_2, v_3\}$ is a separating set which separates the graph into two components; (c) $\{v_1, v_2, v_3\}$ is also a separating set of the sub-component; (d) the resulting tree, B_1, B_2, B_3 are the nodes.

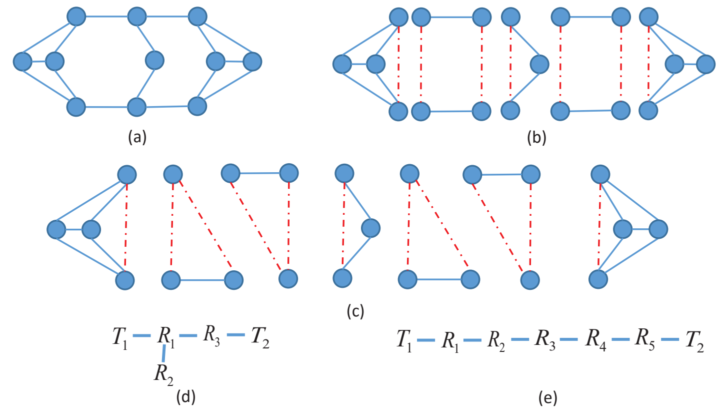


Fig. 2. (a) 2-connected graph, (b) the graph is divided into 3-connected bricks and rings by Tutte decomposition; (c) the graph is divided into 3-connected bricks and triangles; (d)(e) are illustrations of the decomposition tree.

vertex cut $S_1 = \{s_1^{(1)}, s_2^{(1)}, ..., s_k^{(1)}\}$ separating G such that $G[V \setminus S_1]$ into several components. For better visualization, we can consider these components as two components $G[C_1]$ and $G[C_2]$, where $C_1 \cap C_2 = S_1$, and $C_1 \cup C_2 = V$. Next, for $G[C_1]$, if there exists a separating set $S_{21} \subset C_1$ of G, we decompose $G[C_1]$ into $G[C_1^{(1)}]$ and $G[C_1^{(2)}]$, where $C_1^{(1)} \cap C_1^{(2)} = S_{21}$, and $C_1^{(1)} \cup C_1^{(2)} = C_1$; for $G[C_2]$, we apply the same decomposition rule, smaller components $G[C_2^{(1)}]$ and $G[C_2^{(2)}]$ are obtained. The above process can go on, until there exists no separating set $S_i \subset C_j$ of G in each component $G[C_j]$.In other words, $G[C_j]$ cannot be divided into smaller components with the removal of any k nodes in C_j. Finally, when G has been decomposed into components $B_1, B_2, ..., B_p$ with the corresponding separating set $S_1, S_2, ..., S_q$, we can construct a tree whose nodes is the components $B_1, B_2, ..., B_p$, and if the components B_i and B_j are separated by some separating set, there exists an edge between the node B_i and B_j. When G is a 2-connected graph, according to the above definition, G is decomposed into 3-connected components and triangles.

More precisely, we introduce the following $G = (V, E)$ is a k-connected graph, $S_1 = \{s_1^{(1)}, s_2^{(1)}, ..., s_k^{(1)}\}$ is the separating set which separates G into C_1 and C_2, $C_1 \cap C_2 = S_1$. Then we add an virtual edge between any pair $s_i^{(1)}$, $s_j^{(1)}$, $(i, j = 1, 2, ..., k)$ if there exists no edge $(s_i^{(1)}, s_j^{(1)}) \in E$, with the virtual edges, $\widetilde{C_1}$ and $\widetilde{C_2}$ is obtained. Next, if there exists a separating set S_2 of $\widetilde{C_1}$, or S_3 of $\widetilde{C_2}$, the decomposition moves on. Similarly, construct the graphs by adding virtual edges between the nodes in S_2 and S_3. This terminates until there is no separating set in each sub-component.

Fig. 1 (d) illustrates an example of a resulting tree from the decomposition result in Fig. 1 (a).

One important thing to note about this decomposition method is that, the above decomposition is not unique. Though once the separating set is verified, the graph can be decomposed into smaller components. This differs from the Tutte decomposition, which needs to find the split-separator. It is known that after Tutte decomposition is completed, the nodes in different bricks can not constitute a separating set. However, for the above decomposition, this is not always true. e.g.,Fig. 1 shows an example, $\{v_4, v_5, v_6\}$ is a separating set.

IV. Main Results

In this section, we propose our constant factor approximation algorithm for the k-connected m-dominating set problem with any $m \leq k \leq 1$.

A. A new constant factor approximation algorithm for the minimum (k, m)-CDS

In this section, we present a new constant factor approximation algorithm for computing (k, m)-CDS. The rough outline of this algorithm is as follows:

(a) Given an input UDG $G = (V, E)$ which has a feasible solution (and therefore it is k-connected), the algorithm first computes a $(1, m)$-CDS D using an existing algorithm.

(b) For each $i = 2$ to k, we repeat the following steps until $G[D]$ becomes a (k, m)-CDS:

  (i) Recursively decompose $G[D]$ into a set of components (connected subgraphs), each of which does not contain a separating set. In detail, initially, check $G[D]$ has a separating set, and if so, D is split into three subsets; a separating set S, and two nodes subsets D_L and D_R, which are separated by S, i.e. $G[D_L]$ and $G[D_R]$ are disconnected. The same procedure is recursively applied to D_L and D_R and more separating sets inside each of them are identified in a recursive manner.

  (ii) Whenever we find D_L and D_R have no separating set inside during the course of the recursive procedure above, add a bounded number of H-paths (a path from a node in D_L to another node in D_R) with length at most 3 to $G[D]$ so that the separating set can be removed (not separating set anymore).

Algorithm 1 A New Constant Algorithm for Computing (k,m)-CDS, $m \geq k$

Compute $(1,m)$-CDS and set $D \leftarrow (1,m)$-CDS.
for $i = 2$ to k **do**
  if there is a separating set S_1 of $G[D]$ **then**
    repeat
      Decompose $G[D]$ into the (sub-)components.
    until there is no separating set in each (sub-)components.
  end if
  Add H-paths to connect the components, H_1 denotes the intermediate nodes of all the H-paths added.
  if there is a separating set of $G[D \cup H_1]$ **then**
    repeat
      Add an H-path to connect the components.
    until there is no separating set. Let H_2 denote the intermediate nodes of the H-paths added.
  end if
  $D \leftarrow D \cup H_1 \cup H_2$
end for
Return D

(iii) So far, all of the separating sets shared by two components are removed, but there may exist some separating sets between two remote components. Those separating sets are removed by adding a bounded number of H-paths with length at most 3.

(iv) As a result, the connectivity of $G[D]$ is increased by 1.

After the execution of the algorithm, a (k,m)-CDS is returned.

B. Performance Analysis

In this section, we provide theoretical analysis to show Algorithm 1 is in fact a constant factor approximation algorithm for the minimum (k,m)-CDS.

Lemma 1. *After the decomposition of a k-connected graph $G = (V,E)$ is completed, $\mathcal{S} = \{S_1, S_2, ..., S_p\}$ is the corresponding separating set, and $\mathcal{B} = \{B_1, B_2, ..., B_q\}$ is the components. T is a separating set, for any $i = 1,2,...,q$, $T \not\subset B_i$. If T separates G into C_1, C_2,...,C_r, for any C_j, there must exist some $S_i \in \mathcal{S}$ such that $C_j \cap S_i \neq \emptyset$.*

Proof. There are several cases:

- Case 1:$T = T_1 \cup T_2$, $T_1 \cap T_2 = \emptyset$, and $T_1 \subset (B_a \setminus (B_a \cap B_b))$, $T_2 \subset (B_b \setminus (B_a \cap B_b))$. First, we claim that there must exist S_c such that $B_a \cap B_b = S_c$, otherwise, T cannot be a separating set. Since if $B_a \cap B_b \not\subset \mathcal{S}$, according to the decomposition, there must exist at least one B_d between them. It is obvious that $(B_a \setminus T_1)$ and $(B_b \setminus T_2)$ are connected via B_d. By contradiction, assuming that T separates G into C_1, C_2,...,C_r, there exists a component C_j, for any i, $C_j \cap S_i = \emptyset$. Without loss of generality, we assume $C_j \subset (B_a \setminus S_c)$, i.e., for B_a, C_j is only adjacent to T_1. However, after the decomposition completed, the nodes in $B_a \setminus S_c$ is connected to the nodes in $B_b \setminus S_c$ only via S_c. And $|T_1| < k$, this contradicts with the fact that the original graph is k-connected. Fig. 3 (a) helps to understand.

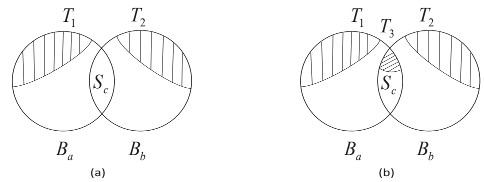


Fig. 3. (a) an illustration of Case 1. (b) an illustration of Case 2.

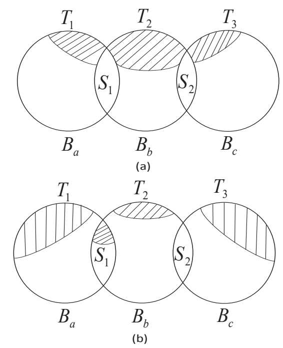


Fig. 4. (a) an illustration of Case 3. (b) an illustration of Case 4.

- Case 2: $T = T_1 \cup T_2 \cup T_3$, $T_i \cap T_j = \emptyset$ $(i,j = 1,2,3)$, and $T_1 \subset (B_a \setminus (B_a \cap B_b))$, $T_2 \subset (B_b \setminus (B_a \cap B_b))$, $B_a \cap B_b = S_c$, $T_3 \subset S_c$. Similarly, based on the decomposition, for B_a, the nodes in $B_a \setminus S_c$ cannot communicate with the nodes in $B_b \setminus S_c$ only by $T_1 \cup T_3$. Otherwise, B_a can be separated by $T_1 \cup T_3$, however, $|T_1| + |T_3| < k$, this is against the the fact that the size of separating set is k.
- Case 3: $T = T_1 \cup T_2 \cup T_3$, $T_i \cap T_j = \emptyset$ $(i,j = 1,2,3)$, and $T_1 \subset B_a$, $T_2 \subset B_b$, $T_3 \subset B_c$, from the above analysis, there must exist $S_1, S_2 \in \mathcal{S}$ such that $B_a \cap B_b = S_1$, $B_b \cap B_c = S_2$. Assuming $T_1 \subset B_a \setminus S_1$, $T_2 \subset B_b \setminus (S_1 \cup S_2)$, and $T_3 \subset B_c \setminus S_2$; the node set $B_1 \setminus (T_1 \cup S_1)$ cannot be only adjacent to T_1, $B_2 \setminus (T_2 \cup S_1 \cup S_2)$ cannot be only adjacent to T_2; the node set $B_3 \setminus (T_3 \cup S_2)$ cannot be only adjacent to T_3. Thus, all the components separated by T has at least one node in some S_i.
- Case 4: $T = T_{11} \cup T_{12} \cup T_2 \cup T_3$, and $T_{11}, T_{12} \subset B_a$, $T_2 \subset B_b$, $T_3 \subset B_c$, $B_a \cap B_b = S_1$, $B_b \cap B_c = S_2$.And $T_{11} \subset B_a \setminus S_1$, $T_{12} \subset S_1$, $T_2 \subset B_b \setminus (S_1 \cup S_2)$, and $T_3 \subset B_c \setminus S_2$; $B_a \setminus (T_{11} \cup T_{12})$ must connect $B_b \setminus S_1$ via $S_1 \setminus T_{12}$, otherwise, $T_{11} \cup T_{12}$ separates B_a, however, $|T_{11}| + |T_{12}| < k$.

If T consists of the nodes from more than 3 components, the proof is similar. □

Lemma 1 is crucial to our performance ratio analysis.

Lemma 2. *After the decomposition is completed, $\mathcal{S} = \{S_1, S_2, ..., S_p\}$ is the corresponding separating set, and $S_i = \{s_1^i, s_2^i, ..., s_k^i\}$ $(i = 1, ..., p)$. If there exists no (s_x^i, s_y^j)-separator where $(i, j = 1, ..p), (x, y = 1, ..k)$, there is no separating set.*

Proof. From Lemma 1, except for the separating set $S_1, S_2, ..., S_p$, all the other separating sets are (s_x^i, s_y^j)-separator. Thus, when we add H-paths in Line 11, at least one pair (s_x^i, s_y^j) cannot be separated any more. □

According to Lemma 2, apparently, the number of the H-paths added in Line 11 can be bounded.

Lemma 3. *For any k-connected graph G, let x be a new vertex which is adjacent to at least $(k+1)$ vertices in G, then the graph G' obtained from G by adding x has no new separating set.*

Since the node added into $C_{k-1,m}$ is at least m-dominated by $C_{k-1,m}$, we have the following

Lemma 4. *The number of the intermediate nodes of the H-path is at most two.*

Theorem 1. *The output $D \cup H_1 \cup H_2$ is k-connected.*

Proof. Line 8 and Line 11 guaranteed the output $D \cup H_1 \cup H_2$ is k-connected. □

Theorem 2. *Algorithm 1 is a constant factor approximation for computing (k, m)-CDS.*

Proof. First, if the (i, m)-CDS D $(i = 1, ..., k-1)$ is decomposed into $B_1, ..., B_q$ with the corresponding separating sets $S_1, ..., S_p$, we have

$$q < |C_{i,m}| = n. \tag{1}$$

By the definition of decomposition,

$$|B_1| + |B_2| + ... + |B_q| = n + p \times i. \tag{2}$$

Since after the decomposition finished, $|B_j| \geq i+1$ for any $j = 1, 2, ..., q$. Thus,

$$(i+1) \times q \leq n + p \times i, \tag{3}$$

i.e., $n + (p-q) \times i \geq q$. Obviously, $q - p = 1$, so we obtained the following $n - i \geq q$.

From Lemma 2 and Lemma 3,

$$|H_1| \leq 2 \times p, |H_2| \leq 2 \times p \times i(i-1)/2, \tag{4}$$

Thus,

$$|C^*_{i+1,m}| \leq |C_{i+1,m}| = |C_{i,m}| + |H_1| + |H_2| \tag{5}$$

$$\leq n + 2n + ni(i-1) = (i^2 - i + 3)n \tag{6}$$

As i varies from 2 to k, each augmentation incurs approximation factor of $O(2^2)$, $O(3^2), \cdots, O(k^2)$, respectively.

As a result, Algorithm 1 has a constant performance ratio of is $O((k!)^2) = O(2^2) \times O(3^2) \times \cdots \times O(k^2)$ with $k \geq 3$. □

V. Structural Complexity Analysis and Comparison

In this section, we study the structural complexity of our algorithm (Algorithm 1) against the existing alternatives by Shi et. al. [6] and Fukunaga [7]. We use Fig 5 to illustrate the main steps of these algorithms. Our algorithm (see Fig 5(a)) first recursively decomposes the $(i, m) - CDS$ (with $i = 1$ as the initial round and i grows up to $k-1$ by 1 in each round) with the corresponding separating sets $\mathcal{S} = \{S_1, S_2, ..., S_p\}$. In this decomposition, we only need to find the separating set arbitrarily in each sub-component (instead of computing all of the separating sets). After we add H-paths to convert these separating sets, we continue to add H-paths to construct $G[D \cup H_1]$ k-connected. Our core strategy is similar to one of the algorithm in [7] (see Fig 5(b)), but is much simpler than that due to the following reasons: The algorithm in [7] finds the minimal separating set in each iteration in a round. This means that the same separating set may be identified many times in each round. The algorithm in [6] first compute m-dominating set D, and then applies an algorithm to compute a k-connected subgraph F on terminal set D. This algorithm is based on an approximation algorithm for computing minimum node-weighted Steiner network, and subset k-connected subgraph. Different from the augmentation problem, this algorithm computes the k-subgraph directly. This algorithm is based on a primal-dual strategy and incurs excessively high complexity and running time compared to the augmentation based strategies.

VI. Conclusion

In this paper, we investigated the problem of constructing highly fault-tolerant virtual backbone, i.e, computing a (k, m)-CDS in wireless networks, where k and m are a pair of integers satisfying $m \geq k \geq 1$. We propose a constant factor polynomial time approximation algorithm to compute (k, m)-CDSs based on decomposition of the graphs. In the future, we will focus on developing approximation algorithm for the minimum k-connected m-dominating set problem with better performance ratio and simpler structure.

Acknowledgement

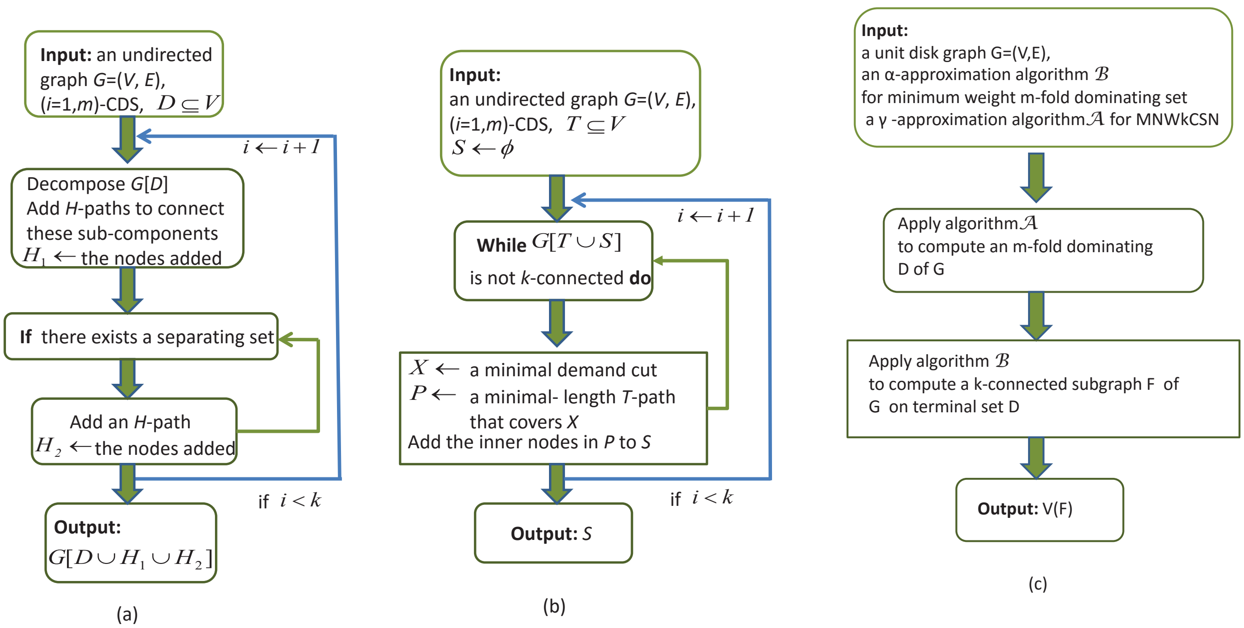
This research was jointly supported by National Natural Science Foundation of China under grants 11471005. This work was also supported by US National Science Foundation (NSF) No. HRD-1345219.

References

[1] Y. Hong, D. Bradley, D. Kim, D. Li, A.O. Tokuta, and Z. Ding, "Construction of Higher Spectral-efficiency Virtual Backbone in Wireless Networks," *Ad Hoc Networks (ADHOC)*, vol. 25, Part A, pp. 228236, February 2015.

[2] Y. Li, D. Kim, F. Zou, and D.-Z. Du, "Constructing Connected Dominating Sets with Bounded Diameters in Wireless Networks," *Proceedings of The 2nd International Conference on Wireless Algorithms, Systems and Applications (WASA 2007)*, August 1-3, 2007, Chicago, IL, USA.

[3] D. Kim, X. Li, F. Zou, Z. Zhang, and W. Wu, "Recyclable Connected Dominating Set for Large Scale Dynamic Wireless Networks," *Proceedings of the 3rd International Conference on Wireless Algorithms, Systems and Applications (WASA 2008)*, October 26-28, 2008, Dallas, TX, USA.



(a) flowchart (b) flowchart

Fig. 5. (a) the flow chart of Algorithm 1; (b) the flow chart of the Simple Algorithm in [7]; (c) the flow chart of the Algorithm in [6].

[4] D. Kim, Y. Wu, Y. Li, F. Zou, and D.-Z. Du, "Constructing Minimum Connected Dominating Sets with Bounded Diameters in Wireless Networks," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 20, no. 2, pp. 147-157, February 2009.

[5] D. Kim, Z. Zhang, X. Li, W. Wang, W. Wu, and D.-Z. Du, "A Better Approximation Algorithm For Computing Connected Dominating Sets in Unit Ball Graphs," *IEEE Transactions on Mobile Computing (TMC)*, vol. 9, no. 8, pp. 1108-1118, August 2010.

[6] Y. Shi, Z. Zhang, and D.-Z. Du, "Approximation Algorithm for Minimum Weight (k, m)-CDS Problem in Unit Disk Graph," available online at `arXiv:1508.05515v1 [cs.DM]`, Aug 2015. (last visited on March 14, 2016)

[7] T. Fukunaga, "Constant-approximation algorithms for highly connected multi-dominating sets in unit disk graphs," `arXiv:1511.09156v1 [cs.DS]`, Nov 2015. (last visited on March 14, 2016)

[8] Z. Nutov, Approximating minimum-cost connectivity problems via uncrossable bifamilies. Acm Transactions on Algorithms, vol. 9, issue 1, pp. 417426, 2012.

[9] W. Wang, B. Liu, D. Kim, D. Li, J. Wang, and Y. Jiang, "A Better Constant Approximation of Minimum 3-connected m-dominating Set Problem in Unit Disk Graph using Tutte Decomposition," *Proceedings of the 34th IEEE International Conference on Computer Communications (INFOCOM 2015)*, April 26-30, 2015, Hong Kong.

[10] Z. Zhang, J. Zhou, Y. Mo, D.-Z. Du, "Performance-Guaranteed Approximation Algorithm for Fault-Tolerant Connected Dominating Set in Wireless Networks," *Proceedings of the 35th IEEE Conference on Computer Communications (INFOCOM 2016)*, April 2016, to appear.

[11] B. Liu, W. Wang, D. Kim, D. Li, J. Wang, A. Tokuta, and Y. Jiang, "On Approximating Minimum 3-connected m-dominating Set Problem in Unit Disk Graph," IEEE/ACM Transactions on Networking, published online. (DOI: 10.1109/TNET.2015.2475335)

[12] D. Kim, W. Wang, X. Li, Z. Zhang, and W. Wu, "A New Constant Factor Approximation for Computing 3-Connected m-Dominating Sets in Homogeneous Wireless Networks," *Proceedings of the 29th IEEE Conference on Computer Communications (INFOCOM 2010)*, March 2010.

[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol 40, pp. 102-114, August 2002.

[14] C.R. Dow, P.J. Lin, S.C. Chen, J.H. Lin,and S.F. Hwang, "A Study of Recent Research Trends and Experimental Guidelines in Mobile Ad Hoc Networks," *Proc. of the 19th International Conference on Advanced Information Networking and Applications (AINA05)*, pp. 72-77, Taiwan, March 2005.

[15] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Proc. of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM99)*, pp.152-162, Washington, USA, August 1999.

[16] A. Ephremides, J. Wieselthier, and D. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proceeding of the IEEE*, vol. 75, issue 1, pp. 56-73, 1987.

[17] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, pp.374-387, April 1998.

[18] B. Das and V. Bharghavan, "Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets," *Proc. of International Conference on Communications (ICC'97)*, pp. 376-380, Montreal, Canada, June 1997.

[19] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," *Proc. of the 3rd ACM Internationl workshop Discrete Algorithm and Methods for Mobile Computing and Communications*, pp. 7-14, Washington, USA, 1999.

[20] M. Thai, F. Wang, D. Liu, S. Zhu, and D.-Z. Du, "Connected dominating sets in wireless networks with different transmission ranges," *IEEE Trans. Mobile Computing*, vol. 6, no. 7, pp. 721-730, 2007.

[21] M. Cardei, X. Cheng, X. Cheng, and D.-Z. Du, "Connected Domination in Multihop Ad Hoc Wireless Networks," *Proc. of International Conference on Computer Science and Informatics*, March 2002.

[22] F. Dai and J. Wu, "On Constructing k-Connected k-Dominating Set inWireless Network," *Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS05)*, 2005.

[23] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," i*Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, New York, USA, June 2002.

[24] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks," *Proc. of the 35th Hawaii International Conference on System Sciences (HICSS'02)*, pp. 297, Hawaii, 2002.

[25] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and K.-I. Ko, "Greedy Approximation for Minimum Connected Dominating Sets," *Theoretical Computer Science*, vol. 329, pp. 325-330, Dec. 2004.

[26] M. Min, F.Wang, D.-Z. Du, and P. M. Pardalos, "Reliable Virtual Backbone Scheme in Mobile Ad Hoc Networks," *Proc. of IEEE International conference on Mobile Ad Hoc and Sensor Systems (MASS'04)*, 2004.

[27] Y. Li, S. Zhu, My T. Thai, and D-Zhu Du, "Localized Construction of Connected Dominating Set in Wireless Networks," NSF International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor and Peerto-Peer Networks (TAWN04), Chicago, June 2004.

[28] X. Cheng, M. Ding, D. Du and X. Jia, "Virtual Backbone Construction in Multihop Ad Hoc Wireless Networks," Wireless Communnications and Mobile Computing, vol. 6, issue 2, pp. 183-190, 2006.

[29] K. Mohammed, L. Gewali, and V. Muthukumar, "Generating Quality Dominating Sets for Sensor Network," *Proc. of the 6th International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05)*, pp. 204-211, Nevada, USA, August 2005.

[30] Y. Zeng, X. Jia, and Y. He, "Energy Efficient Distributed Connected Dominating Sets Construction in Wireless Sensor Networks," *Proc. of International Wireless Communications and Mobile Computing Conference (IWCMC'06)*, pp. 797-802, Vancouver, Canada, July 2006.

[31] S. Funke, A. Kesselman, U. Meyer, and M. Segal, "Simple Improved Distributed Algorithm for Minimum CDS in Unit Disk Graphs," *ACM Transactions on Sensor Network*, vol. 2, no. 3, pp. 444-453, August 2006.

[32] R. Diestel,Graph Theory 3rd ed, "Graduate Texts in Mathematics," vol.173, Springer-Verlag, Heidelberg, 2005.

[33] M.T. Thai, N. Zhang, R. Tiwari, and X. Xu, "On Approximation Algorithms of k-Connected m-Dominating Sets in Disk Graphs," Theoretical Computer Science, vol. 358, pp. 49-59, 2007.

[34] W. Shang, F. Yao, P. Wan, and X. Hu, "On Minimum m-Connected k-Dominating Set Problem in Unit Disc Graphs," *Journal of Combinatorial Optimization*, Dec. 2007.

[35] Y. Shi, Y. Zhang, Z. Zhang and W. Wu, "A Greedy Algorithm for the Minimum 2-connected m-fold Dominating Set Problem," *Journal Of Combinatorial Optimization (JOCO)*,DOI 10.1007/s10878-014-9720-6, 2014.

[36] F. Wang, M. T. Thai, and D.-Z. Du, "2-Connected Virtual Backbone in Wireless Network," *IEEE Transactions on Wireless Communications*, vol.8, no. 3, pp. 1230-1237, Mar. 2009.

[37] W.T. Tutte, "Connectivity in Graphs," Oxford U.P, 1966.

[38] D. B. West, "Introduction to Graph Theory (Second Edition)," Prentice-Hall, 2001.

[39] Y. Wu, F. Wang, M. T. Thai and Y. Li, "Constructing k-Connected m-Dominating Sets in Wireless Sensor Networks," *Military Communications Conference (MILCOM'07)*, Orlando, FL, October 29-31, 2007.

[40] Y. Wu and Y. Li, "Construction Algorithms for k-Connected m-Dominating Sets in Wireless Sensor Networks," *Proc. of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc'08)*, Hong Kong, China, May 26-30, 2008.

[41] W. Wang, D. Kim, M. An, W. Gao, X. Li, Z. Zhang, W. Wu, "On Construction of Quality Fault-Tolerant Virtual Backbone in Wireless Networks," *IEEE/ACM Transactions on Networking (ToN)*, pp. 1499-1510, 2013.

[42] W. Holberg, "The decomposition of graphs into k-connected components," *Discrete Mathematics*, vol. 109, pp. 133145, 1992.

[43] T. Shi, S. Cheng, Z. Cai, and J. Li, "Adaptive Connected Dominating Set Discovering Algorithm in Energy-Harvest Sensor Networks," *Proceeding of The 35rd Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*, San Francisco, CA, April 10-15, 2016, San Francisco, CA, USA.

[44] S. Cheng, Z. Cai, J. Li, and X. Fang, "Drawing Dominant Dataset from Big Sensory Data in Wireless Sensor Networks," *Proceeding of The 34rd Annual IEEE International Conference on Computer Communications (INFOCOM 2015)*, April 26-30, 2015, Hong Kong, Hong Kong.

[45] J. Gao, J. Li, Z. Cai, and H. Gao, "Composite Event Coverage in Wireless Sensor Networks with Heterogeneous Sensors," *Proceeding of The 34rd Annual IEEE International Conference on Computer Communications (INFOCOM 2015)*, April 26-30, 2015, Hong Kong, Hong Kong.