Connected Dominating Sets in Wireless Networks with Different Transmission Ranges

My T. Thai, *Member*, *IEEE*, Feng Wang, *Member*, *IEEE*, Dan Liu, *Student Member*, *IEEE*, Shiwei Zhu, *Student Member*, *IEEE*, and Ding-Zhu Du, *Member*, *IEEE*

Abstract—Since there is no fixed infrastructure or centralized management in wireless ad hoc networks, a Connected Dominating Set (CDS) has been proposed to serve as a virtual backbone. The CDS of a graph representing a network has a significant impact on the efficient design of routing protocols in wireless networks. This problem has been studied extensively in Unit Disk Graphs (UDG), in which all nodes have the same transmission ranges. However, in practice, the transmission ranges of all nodes are not necessarily equal. In this paper, we model a network as a disk graph and introduce the CDS problem in disk graphs. We present two efficient approximation algorithms to obtain a minimum CDS. The performance ratio of these algorithms is constant if the ratio of the maximum transmission range over the minimum transmission range in the network is bounded. These algorithms can be implemented as distributed algorithms. Furthermore, we show a size relationship between a maximal independent set and a CDS as well as a bound of the maximum number of independent neighbors of a node in disk graphs. The theoretical analysis and simulation results are also presented to verify our approaches.

Index Terms—Connected dominating set, independent set, disk graph, wireless network, virtual backbone.

1 Introduction

In wireless ad hoc networks, there is no fixed or predefined infrastructure. Nodes in wireless networks communicate via a shared medium, through either a single-hop communication or multihop relays. Although there is no physical backbone infrastructure, a virtual backbone can be formed by constructing a Connected Dominating Set (CDS). Given an *undirected* graph $G = (V, E)$, a subset $C \subseteq V$ is a CDS of G if, for each node $u \in V$, u is either in C or there exists a node $v \in C$ such that $uv \in E$ and the subgraph induced by C, i.e., $G(C)$, is connected. The nodes in the CDS are called *dominators* and other nodes are called *dominatees*. With the help of the CDS, routing is easier and can adapt quickly to network topology changes. To reduce the traffic during communication and simplify the connectivity management, it is desirable to construct a Minimum CDS (MCDS).

The CDS problem has been studied intensively in Unit Disk Graphs (UDG) [1], in which all nodes have the same transmission range. The MCDS problem in UDG has been shown to be NP-hard [13]. To build a CDS, most of the current algorithms first find a Maximal Independent Set (MIS) I of G and then connect all nodes in I to form a CDS. The independent set I is a subset of V such that, for any two nodes $u, v \in I$, $uv \notin E$. A maximal independent set is an independent set into which no more nodes can be added to retain the nonadjacency property. The most relevant works using this scheme are in [2], [3], and [8].

However, in practice, the transmission ranges of all nodes are not necessarily equal. In this case, a wireless ad hoc network can be modeled using a directed graph $G = (V, E)$. The nodes in V are located in the two-dimensional euclidean plane and each node $v_i \in V$ has a transmission range $r_i \in [r_{min}, r_{max}]$. A directed edge $(v_i, v_j) \in E$ if and only if $d(v_i, v_j) \le r_i$, where $d(v_i, v_j)$ denotes the euclidean distance between v_i and v_j. Such graphs are called *Disk Graphs* (DG). An edge (v_i, v_j) is bidirectional if both (v_i, v_j) and (v_j, v_i) are in E, i.e., $d(v_i, v_j) \le min\{r_i, r_j\}$. In this paper, we only study the CDS problem in disk graphs where all the edges in the network are bidirectional, called Disk Graphs with Bidirectional links (DGB). In this case, G is undirected. Fig. 1 gives an example of DGB representing a network. In Fig. 1, the dotted circles represent the transmission ranges and the black nodes represent a CDS.

While the study of CDS with UDG in homogeneous networks has drawn a lot of attention, the study of CDS with DG in heterogeneous networks, which is even harder, has been insufficient. In this paper, we present two constant approximation algorithms for computing a minimum CDS in DGB where the ratio of the maximum transmission range over the minimum transmission range, called the *transmission range ratio*, is bounded. We first introduce the centralized versions and later show how to implement them as distributed algorithms. The contributions of this paper are as follows:

- *M.T. Thai is with the Department of Computer and Information Science and Engineering, University of Florida, CSE Building, Room E301, PO Box 116120, Gainesville, FL, 32611-6120. E-mail: mythai@cise.ufl.edu.*
- *F. Wang, D. Liu and S. Zhu are with the Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CS Building, 200 Union Street SE, Minneapolis, MN 55455. E-mail: {fwang, danliu, zhu}@cs.umn.edu.*
- *D.-Z. Du is with the Department of Computer Science and Engineering, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, PO Box 830688, MS EC31, Richardson, TX 75083-0688. E-mail: dzdu@utdallas.edu.*

Manuscript received 24 Sept. 2005; revised 12 Apr. 2006; accepted 17 Oct. 2006; published online 7 Feb. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0284-0905. Digital Object Identifier no. 10.1109/TMC.2007.1034.

1536-1233/07/$25.00 © 2007 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

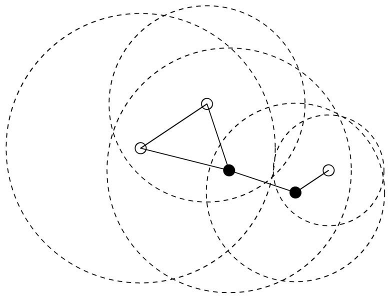


Fig. 1. A disk graph with bidirectional links.

1. We provide a size relationship between an MIS and a CDS in DGB. We also show an upper bound of the number of independent neighbors of any node in DGB.
2. We present two efficient algorithms to solve the CDS problem in DGB. We also provide the theoretical analysis of these algorithms. Their performance ratios are constant when the transmission range ratio is bounded. Furthermore, we illustrate how to implement these algorithms as the distributed algorithms.
3. We analyze the performance of our algorithms through simulations and compare it to that of the algorithm proposed in [2].

The remainder of this paper is structured as follows: Section 2 describes the related research work on the CDS problem, mainly focused on UDG. The size relationship between an MIS and a CDS in DGB is shown in Section 3. The two algorithms and their theoretical analyses are discussed in Section 4. Section 5 presents the performance comparison of the algorithm proposed in [2] and ours through simulations. The distributed implementations are illustrated in Section 6 and Section 7 ends the paper with conclusions and future work.

2 Related Work

In this section, we describe the main ideas of many related works on constructing a CDS with their theoretical analysis results. For experimental evidence of their performances, readers are referred to [21]. In [21], Basagni et al. present excellent ns2-based performance comparisons of many leading protocols in this area. They also assess the effect of "degree of localization" on protocol duration, energy consumption, message overhead, route length, and CDS size.

Algorithms on constructing a CDS can be divided into two categories based on their algorithm designs: *centralized* algorithms and *decentralized* algorithms. The *centralized* algorithms usually yield a CDS with a better performance ratio than that of decentralized algorithms. The decentralized algorithms can be further divided into two categories: *distributed* algorithms and *localized* algorithms. In the distributed algorithms, the decision process is decentralized and serialized. In the localized algorithms, the decision process is not only distributed, but also requires only a *constant* number of communication rounds. Based on the network models, these algorithms can be classified into two types: *undirected* graphs and *directed* graphs. For undirected graphs, we can further divide them into two categories: *general graphs* and *unit disk graphs*. When modeling a network as a general undirected graph G, the algorithm's performance ratio usually relates to Δ where Δ is the maximum degree of G. When modeling a network as a unit disk graph, the performance ratio is usually constant due to the special geometric structure of UDG.

2.1 Undirected Graphs

In general graphs, several works have been studied in recent research literature. In [4], Guha and Khuller proposed two polynomial time algorithms to construct a CDS in a general undirected graph G. These algorithms are greedy and centralized. The first one has a performance ratio of $2(H(\Delta)+1)$ where H is a harmonic function. The idea of this algorithm is to build a spanning tree T rooted at the node with a maximum degree and grow T until all nodes are added to T. The nonleaf nodes in T form a CDS. In particular, all nodes in a given network are marked white initially. The greedy function that the algorithm uses to add nodes into T is the number of the white neighbors of each node or a pair of nodes. The one with the largest such number is marked in black and its neighbors are marked in gray. These nodes (black and gray nodes) are then added to T. The algorithm stops when no white node exists in G. The second algorithm is an improvement of the first one. This algorithm consists of two phases. The first phase is to construct a dominating set and the second phase is to connect the dominating set using a Steiner tree algorithm. With such improvement, the second algorithm has a performance factor of $H(\Delta)+2$. These algorithms later were studied and implemented by Das et al. [10], [11], [12]. In [5], Ruan et al. introduced another centralized and greedy algorithm of which the performance ratio is $(2+\ln\Delta)$.

For the localized algorithms, Wu and Li [6] proposed a simple algorithm that can quickly determine a CDS based on the connectivity information within the 2-hop neighborhood. This approach uses a marking process. In particular, each node is marked true if it has two unconnected neighbors. All the marked nodes form a CDS. The authors also introduced some dominant pruning rules to reduce the size of the CDS. In [2], Wan et al. showed that the performance ratio of [6] is within a factor of $O(n)$, where n is the number of nodes in a network.

2.1.1 Unit Disk Graphs

In the UDG, most proposed algorithms are distributed algorithms where the main approach is to find a Maximal Independent Set (MIS) and then connect this set. Note that, in undirected graphs, an MIS is also a Dominating Set (DS). In [2], [14], [15], the authors proposed a distributed algorithm for a CDS problem in UDG. This algorithm consists of two phases and has a constant performance ratio of 8. The algorithm first constructs a spanning tree. Then, each node in a tree is examined to find an MIS for the first phase. All nodes in an MIS are colored black. In the second phase, more nodes are added (colored blue) to connect those black nodes. Later, Cardei et al. presented another 2-phase distributed algorithm for a CDS in UDG

[3]. This algorithm has the same performance ratio of 8. However, the improvement over [2] is the message complexity. The root does not need to wait for the COMPLETE message from the furthest nodes. We can apply the algorithms proposed in [2], [3] to DGB to construct a CDS. However, this simple extension will not result in a good performance ratio as shown in Section 4. Recently, Li et al. proposed another distributed algorithm with a better approximation ratio, which is $(4.8 + \ln 5)$ [8]. This algorithm also has two phases. In the first phase, an MIS is found. In the second phase, a Steiner tree algorithm is used to connect the MIS.

For the localized algorithms, in [9], Alzoubi et al. proposed a localized 2-phase algorithm with a performance ratio of 192. In the first phase, an MIS is constructed using the one-hop neighbors information. Specifically, once a node knows that it has the smallest ID among its neighbors, it becomes a dominator. In the second phase, the dominators are responsible for identifying a path to connect the MIS. In [7], Li et al. proposed another localized algorithm with a performance ratio of 172. This localized algorithm has only one phase. A node marks itself as a dominator if it can cover the highest number of white nodes compared to its 2-hop neighbors.

2.2 Directed Graphs

In *directed* graphs, a CDS C must meet the following two requirements: 1) C is strongly connected and 2) for any node $v \notin C$, there exists a directed path from v to at least one node in C. Such a set is referred as Strongly CDS (SCDS). In [17], Wu extended the marking process, previously proposed in [6], to networks with unidirectional links to find an SCDS. Later, Dai and Wu generalized their pruning rules used in [6] and [17] to be applicable to any number of neighbors in a directed graph [18]. This generalized pruning rule, called Rule k, does not guarantee a constant approximation ratio. Instead, the authors showed a "probabilistic performance ratio." In UDG, the *average* size of the dominating size derived from Rule k was proved to be upper bounded by a constant. In DG, this claim is held if the *nonrestricted* Rule k, which requires *global* information, is applied. Thus, in the DG context, the proposed pruning rule is no longer localized.

Most of the constant approximation algorithms are for the CDS problem in UDG. However, in practice, the transmission ranges of the nodes in a network are not necessarily equal. Such a network can be modeled as a disk graph. In this paper, we present two constant approximation algorithms for the CDS problem in DGB with a bounded transmission range ratio. The main approach is to construct an MIS and then connect them. Thus, we need to analyze the size relationship between a CDS and an MIS, which is shown in the next section.

3 The Size Relationship Between a CDS and a Maximal Independent Set

In this section, we prove the size relationship between any MIS and a CDS of a given DGB. Let us first introduce our notations, which will be used throughout this paper.

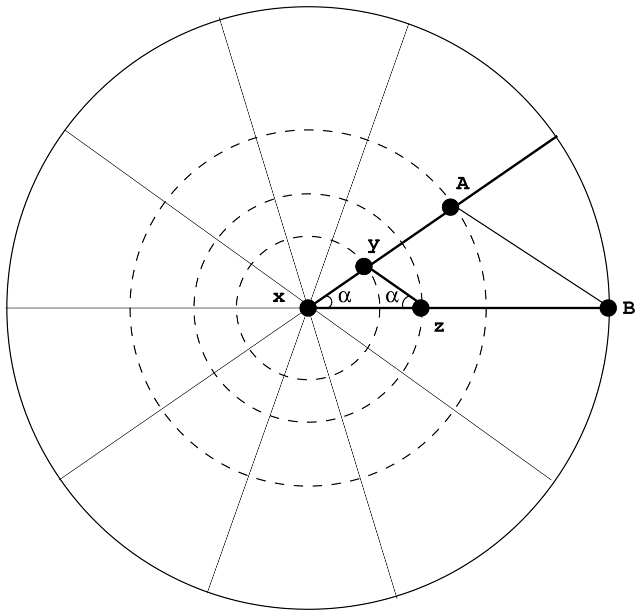


Fig. 2. On the proof of maximum number of independent neighbors.

- $k = \frac{r_{max}}{r_{min}}$.
- $$K = \begin{cases} 5 & \text{if } k = 1 \\ 10\lfloor \frac{\ln k}{\ln(2\cos(\pi/5))} \rfloor & \text{otherwise.} \end{cases}$$
- OPT: an optimal solution of the CDS problem in DGB.
- opt: the size of OPT.
- $N_{ID}(u)$: the independent neighbors of node u.

Fact 1. *Given a DGB $G = (V, E)$ and three nodes x, y, and z such that $xy \in E$, $xz \in E$, $d(x, y) \le d(x, z)$, and $d(y, z) \le d(x, y)$, then y and z are adjacent.*

Proof. The disk at y has radius at least $d(x, y)$ and the disk at z has radius at least $d(x, z)$. Therefore, both disks have radius at least $d(y, z)$. Hence, y and z are adjacent. □

Lemma 1. *In a DGB, the size of $N_{ID}(u)$ is bounded by K, i.e., $|N_{ID}(u)| \le K$.*

Proof. When $k = 1$, a DGB is a UDG. Thus, the lemma holds.

When $k > 1$, consider a node x and all nodes that are adjacent to x. Without loss of generality, assume that the disk at x has radius 1. Then, a node y that is adjacent to x has radius at least $d(x, y)$ and at most k. Thus, all nodes that are adjacent to x lie within the circle at center x with radius k.

We first evenly divide this area into several small areas (sectors) A_i with rays (half lines) at x. Fig. 2 shows 10 areas A_i. Two adjacent rays form an angle α. Suppose xy and xz are two rays with angle α between them. Suppose $d(x, y) \le d(x, z)$ and $d(y, z) = d(x, y)$. Then, from Fact 1, we know that y and z are adjacent. Since $d(y, z) = d(x, y)$, we have

$$\angle xzy = \alpha.$$

Therefore,

$$d(x,z)/d(x,y) = 2\cos\alpha.$$

Now, we further divide each area A_i into subareas by circles at x with radius $1, 2\cos\alpha, (2\cos\alpha)^2, \ldots, (2\cos\alpha)^j$. Note that

$$(2\cos\alpha)^j \leq k.$$

Hence,

$$j = \left\lfloor \frac{\ln k}{\ln(2\cos\alpha)} \right\rfloor.$$

We claim that all nodes which are adjacent to x and lie in each subarea form a clique. Indeed, let A and B be such nodes as shown in Fig. 2. Then, x, A, and B satisfy the condition in Fact 1.

Therefore, there are at most $\lfloor \frac{\ln k}{\ln(2cos\alpha)} \rfloor (2\pi/\alpha)$ subareas. In other words, x can be adjacent to at most $\lfloor \frac{\ln k}{\ln(2cos\alpha)} \rfloor (2\pi/\alpha)$ independent nodes.

Let $f(\alpha) = \lfloor \frac{\ln k}{\ln(2cos\alpha)} \rfloor (2\pi/\alpha)$. Note that, in our proof, $\alpha = 2\pi/m$, where m is an integer and $m > 6$. We thus need to find a local minima of $f(\alpha)$, where $0 < \alpha < 2\pi/6$. With some algebraic steps, we have $\alpha = 2\pi/10$. Hence, when $k > 1$, x can be adjacent to at most $10\lfloor \frac{\ln k}{\ln(2cos(\pi/5))} \rfloor$ independent nodes. □

Theorem 1. *In a DGB $G = (V, E)$, the size of any maximal independent set is upper bounded by $Kopt$ where $k = \frac{r_{max}}{r_{min}}$ and $K = 5$ if $k = 1$; otherwise, $K = 10\lfloor \frac{\ln k}{\ln(2cos(\pi/5))} \rfloor$.*

Proof. Let I be an MIS. By Lemma 1, no node in OPT can dominate more than K nodes in I. Thus, the theorem follows: $|I| \leq Kopt$. □

4 Approximation Algorithms and Analysis

In this section, we present two constant approximation algorithms for the CDS problem in DGB with bounded transmission range ratio k and analyze their performance ratios.

4.1 First Algorithm

One simple way to approximate a CDS in DGB is to apply the algorithm proposed in [2], hereafter referred to as Wan's algorithm. In the DGB context, this algorithm has a performance ratio of $2K$. Recall that Wan's algorithm has two phases. In the first phase, a spanning tree is constructed in order to form an MIS I. Note that all nodes in I are colored black and other nodes are colored gray. In the second phase, all black nodes are connected in the following way: Connect two disconnected black nodes u and v (assume that $level(u) < level(v)$ in the spanning tree) by finding a gray node w that is a parent of v in T and a neighbor of u in G and color this node blue.

However, in this section, we present a better way to connect I. Our main idea is to connect all nodes in I by using a Steiner tree, which is a tree interconnecting all nodes in I. The nodes in the Steiner tree that are not in I are called Steiner nodes. To reduce the size of an obtained CDS, we need to find a Steiner tree with the Minimum number of Steiner Nodes (MSN). We can define this problem as follows:

Definition 1. *Steiner Tree with MSN (ST-MSN). Given a graph $G = (V, E)$ and a set of nodes $V' \subseteq V$ called* terminals, *construct a Steiner tree T that connects all the terminals such that the number of Steiner nodes is minimum.*

The ST-MSN problem has not been studied much in DGB, unlike its geometric version in the euclidean plane, which has been studied extensively [19], [20]. Unfortunately, some results cannot be extended to DGB. For example, two points with distance 2 can be connected with a Steiner point in the euclidean plane; however, two nodes with distance 2 may not be able to be connected by a Steiner node since such a node may not exist. In addition, the proof of an ST-MSN in DGB is quite different from the proof of an ST-MSN in the euclidean plane, which also becomes a fundamental part in our first algorithm.

4.1.1 Algorithm Description

The First Algorithm (TFA) has two phases. The first phase is to find an MIS I that satisfies the following lemma:

Lemma 2. *Any pair of complementary subsets of a constructed MIS has a distance of exactly two hops.*

Note that this condition is vital for the second phase to work. Since an obtained MIS I from Wan's algorithm [2] satisfies this condition, we can use that procedure to find I. In the second phase, we construct an ST-MSN to interconnect all nodes in I as follows: Define a *black-blue component* as a connected component of the subgraph induced only by black and blue nodes, *ignoring connections between blue nodes*. Initially, we have $|I|$ black-blue components. Let B be a set of Steiner nodes, called blue nodes. Initially, B is empty. From Lemma 1, we know that each node is adjacent to at most K independent nodes. In other words, a blue node is adjacent to at most K black nodes. Color all nodes in $V - I$ gray. At each iteration, we can find a gray node that is adjacent to the most number of black-blue components and color it blue. Formally, for j from K to 2, at each iteration j, find a gray node v such that v is adjacent to at least j black nodes in *different* black-blue components. Color v blue and recompute the black-blue components as described in Algorithm 1.

Algorithm 1 The First Algorithm (TFA)

INPUT: A DGB $G = (V, E)$, all nodes are white
OUTPUT: A CDS of G
$I = \emptyset$; $B = \emptyset$
Use the procedure in [2] to compute I. At this stage, all nodes in I are black and other nodes are gray.
for $j = K$ to 2 **do**
  while There exists a gray node v adjacent to at least j black nodes in different black-blue components **do**
    Color v blue
    $B = B \cup \{v\}$
  end while
end for
Return $I \cup B$

4.1.2 Theoretical Analysis

The CDS in this algorithm is a union of set I and set B. To analyze the performance ratio of our algorithm, we first compare the size of set B to opt. Recall that B is a set of all the Steiner nodes. Let T^* be an optimal tree when connecting a given set I and $C(T^*)$ be the number of the Steiner nodes in T^*. We have this following lemma:

Lemma 3. *The size of B obtained from TFA is at most $(2 + \ln K)C(T^*)$.*

Proof. Let $n = |I|$ and $p = |B|$. If $n = 1$, then the lemma is trivial. Assume that $n \geq 2$, thus, $C(T^*) \geq 1$. Let $v_j, j = 1 \dots p$ be the blue nodes in the order of appearance in the second phase. Let a_i be the number of the black-blue components after $v_1, \dots, v_i$ turns blue. Since every black-blue component contains a black node which is adjacent to a Steiner node of T^*, there exists a node v_i which is adjacent to at least $\frac{a_i}{C(T^*)}$ black-blue components. Thus, we have

$$a_{i+1} \leq a_i - \frac{a_i}{C(T^*)} + 1.$$

Hence, we have this following recurrence:

$$\begin{aligned} a_i &\leq a_{i-1} - \frac{a_{i-1}}{C(T^*)} + 1 \\ &\leq a_{i-1}\left(1 - \frac{1}{C(T^*)}\right) + 1 \\ &\leq a_{i-2}\left(1 - \frac{1}{C(T^*)}\right)^2 + \left(1 - \frac{1}{C(T^*)}\right) + 1 \\ &\leq \dots \\ &\leq a_0\left(1 - \frac{1}{C(T^*)}\right)^i + \sum_{j=0}^{i-1}\left(1 - \frac{1}{C(T^*)}\right)^j \\ &\leq a_0\left(1 - \frac{1}{C(T^*)}\right)^i + C(T^*). \end{aligned}$$

For the last step in the above recurrence, we note that the second term $\sum_{j=0}^{i-1}(1 - \frac{1}{C(T^*)})^j$ is the geometric series and it will converge to $C(T^*)$. After $i = C(T^*) \ln \frac{a_0}{C(T^*)}$ iterations, the number of black-blue components will be

$$\begin{aligned} a_i &\leq a_0\left(1 - \frac{1}{C(T^*)}\right)^i + C(T^*) \\ &\leq e^{-\frac{i}{C(T^*)}} + C(T^*) \\ &\leq 2C(T^*). \end{aligned}$$

Therefore, the total number of blue nodes is bounded as follows:

$$\begin{aligned} |B| &\leq i + 2C(T^*) \leq C(T^*)(\ln \frac{a_0}{C(T^*)} + 2) \\ &\leq C(T^*)(\ln \frac{n}{C(T^*)} + 2) \leq (2 + \ln K)C(T^*). \end{aligned}$$

□

Theorem 2. *The first algorithm produces a CDS with the size bounded by $(K + 2 + \ln K)opt$.*

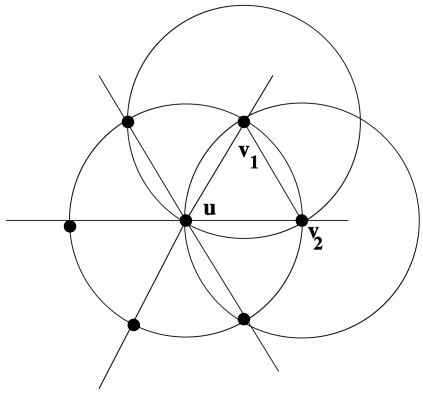


Fig. 3. On the proof of five independent neighbors.

Proof. From Theorem 1 and Lemma 3, we have

$$|CDS| = |I| + |B| \leq (K + 2 + \ln K)opt.$$

□

Corollary 1. *If the transmission range ratio k is bounded, then TFA has an approximation factor of $O(1)$.*

4.2 Second Algorithm

In the previous proposed algorithm, we find an MIS based on the spanning tree which is constructed based on the connectivity information of a given network. In this section, we show an effect of the size of the disks on the size of an MIS.

Lemma 4. *In a DGB G, there exists a node that is adjacent to at most five independent nodes.*

Proof. Let D be a disk with radius r_{min} centered at node u. We prove that u has at most five independent neighbors by contradiction. Suppose that u has more than five independent neighbors. Let v_j, $1 \leq j \leq 6$, be the independent neighbors of u. Then, there exist two nodes that lie in a sector with the angle less than or equal to 60 degrees. Without loss of generality, assume that v_1 and v_2 are such nodes as shown in Fig. 3. Then, $d(v_1, v_2) \leq r_{min}$. Hence, v_1 and v_2 are connected, contradicting to our assumption. □

Note that the subgraph of a DGB is still a DGB. Hence, let us consider the algorithm to find an MIS as shown in Algorithm 2.

Algorithm 2 Choose Smallest Disks
1: INPUT: A DGB $G = (V, E)$
2: OUTPUT: A Maximal Independent Set I
3: $I = \emptyset$
4: **while** $V \neq \emptyset$ **do**
5: Find a node $u \in V$ with the smallest radius, color u black
6: $I = I \cup \{u\}$
7: $V = V - \{u\} - N(u)$
8: **end while**
9: Return I

In this algorithm, at each iteration, we find a node with the smallest radius in V, color it black, and remove this node and its neighbors from V. This step runs iteratively until V is empty. The black nodes form an MIS I. Let I^* be an optimal MIS of G, i.e., $|I^*| \geq |I|$ for any MIS I, we have:

Lemma 5. *The size of I is at least $\frac{|I^*|}{5}$.*

Proof. Every node $v \in V$ is either in I or adjacent to some nodes in I. Since $I^* \subset V$, every node $v \in I^*$ is either in I or adjacent to some nodes in I. Let $N[u]$ be the closed neighbors of u when adding u into I, i.e., $N[u] = N(u) \cup \{u\}$. Then, every node $v \in I^*$ is in $N[u]$ for some $u \in I$. Because, at each step, we choose a node u with the smallest disk, each u has at most five independent nodes (Lemma 4). Thus, each $N[u]$ contains at most five vertices from I^*. This results in $|I| \geq \frac{|I^*|}{5}$. □

Now, let us color the biggest disks instead of the smallest disks black. Specifically, as shown in Algorithm 3, at each iteration, we find a node with the largest transmission range in V and color it black. We remove this node and its neighbors from V. The set of black nodes forms a maximal independent set I.

Algorithm 3 Choose Biggest Disks
```
INPUT: A DGB G = (V, E)
OUTPUT: A Maximal Independent Set I
I = ∅
while V ≠ ∅ do
  Find a node u ∈ V with the biggest radius, color u black
  I = I ∪ {u}
  V = V − {u} − N(u)
end while
Return I
```

Again, let I^* be an optimal MIS of G, and we have the following lemma:

Lemma 6. *The size of I is at least $\frac{|I^*|}{K}$.*

Proof. Using the same approach in the previous proof, by Lemma 1, each $N[v]$ contains at most K independent nodes in I^*. This follows that $|I| \geq \frac{|I^*|}{K}$. □

We believe that the size of I obtained from Algorithm 3 is less than that obtained from The First Algorithm (TFA) due to the above lemma. Thus, we introduce The Second Algorithm (TSA) and its performance is evaluated by simulations. In this algorithm, we first find an MIS I using Algorithm 3. Then, we connect I by choosing a node that is adjacent to the highest number of black-blue components and color it blue. The details of TSA are shown in Algorithm 4.

Algorithm 4 The Second Algorithm (TSA)
```
INPUT: A DGB G = (V, E), all nodes are white
OUTPUT: A CDS
I = ∅; B = ∅
I = ChooseBiggestDisks(G)
while G(I) is disconnected do
  Select a white node u such that u is adjacent to the
  highest number of black-blue components
  Color u blue
  B = B ∪ {u}
end while
Return I ∪ B
```

5 Simulation Results

In the previous section, we evaluated our algorithms through theoretical analysis. In this section, we conduct the simulation experiments to compare the performance (in terms of the size of CDS) of three algorithms: Wan's Algorithm [2] (WA), The First Algorithm (TFA), and The Second Algorithm (TSA). Recall that the improvement of TSA over WA is that we use the Steiner tree with the minimum number of Steiner nodes to interconnect all black nodes. The improvement of TSA over TFA is that we select nodes with largest transmission ranges as the black nodes. Moreover, we are interested in comparing the size of the black nodes obtained from each algorithm to see whether the approach of choosing the biggest disks can result in the smallest number of black nodes. Since the number of black nodes in WA and TFA are the same, let I_1 denote the size of black nodes obtained from either WA or TFA. Let I_b denote the size of black nodes obtained from TSA and I_s be the size of black nodes obtained from the Choose Smallest Disks (CSD) algorithm. We study three network parameters that may affect the algorithm performance:

1. n, the number of nodes in a given network,
2. k, the ratio of the largest transmission range to the smallest transmission range, i.e., $k = \frac{r_{max}}{r_{min}}$, and
3. the network density, i.e., the number of nodes per area

5.1 Effects of Number of Nodes

To evaluate the performance of the three proposed algorithms under different numbers of nodes, we randomly deployed n nodes to a fixed area of 800 m x 800 m. n changed from 10 to 200 with an increment of 1. Each node v_i randomly chose the transmission range $r_i \in [r_{max}, r_{min}]$, where $r_{max} = 600$ m and $r_{min} = 200$ m. For each value of n, 1,000 network instances were investigated and the results were averaged.

As can be seen in Fig. 4a, the size of a CDS obtained from TSA is the smallest among all three algorithms. Specifically, the size of the CDS obtained from TSA is 3.3 percent smaller than that of TFA and 9.1 percent smaller than that of WA. The results indicate that constructing the Steiner tree with the minimum number of Steiner nodes to interconnect the maximal independent set can reduce the size of the CDS. In addition, choosing the biggest disk as a black node can reduce the size of the CDS as well.

Fig. 4b shows the comparison of the number of black nodes obtained from TFA, CSD, and TSA. The number of black nodes I_b obtained from TSA is smaller than that of TFA. The Choose Smallest Disks (CSD) algorithm returns the biggest number of black nodes I_s as shown in Fig. 4b. This is consistent with our expectations as we have analyzed in the previous section.

Fig. 4 also shows how the number of nodes in a network affects the size of the CDS. In particular, the size of the CDS increases as the number of nodes increases. This is because the number of nodes that need to be dominated is larger when we deploy more nodes.

5.2 Effects of the Transmission Range Ratio

We also conducted simulations to compare the performance of all three algorithms when changing the transmission

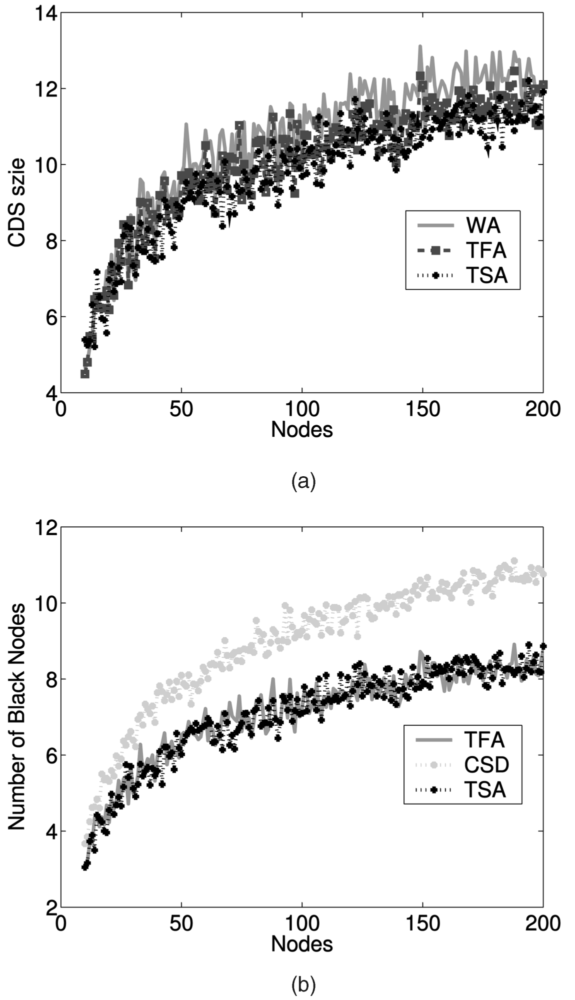


Fig. 4. Effects of number of nodes. (a) Compare the CDS size. (b) Compare the MIS size.

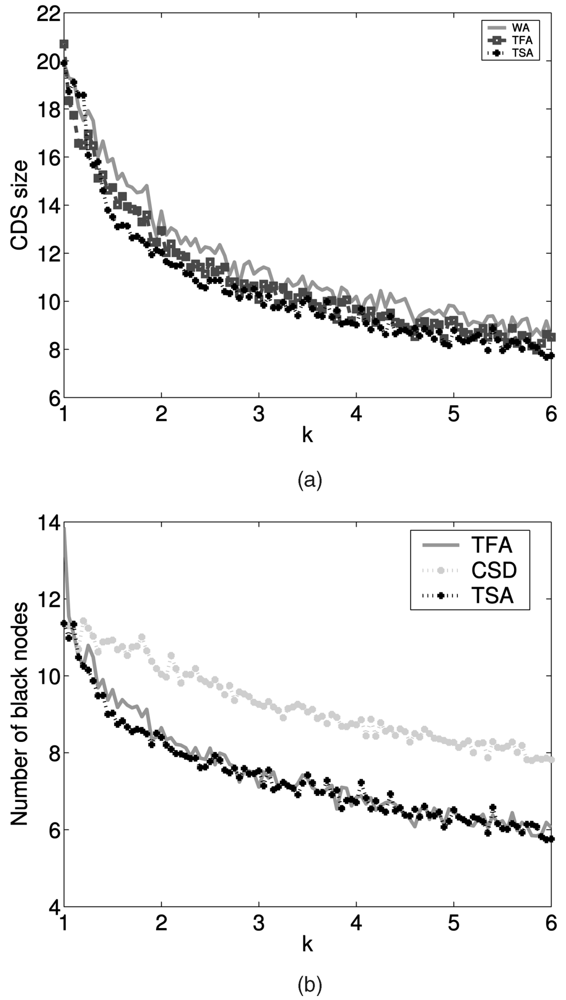


Fig. 5. Effects of the transmission range ratio. (a) Compare the CDS size. (b) Compare the MIS size.

range ratio k as well as to see how this change affects the size of an obtained CDS. To change k, we fixed $r_{min} = 200$ m and changed r_{max} from 200 m to 1,200 m with an increment of 10. In this experiment, we randomly deployed 100 nodes into a fixed area of size 800 m x 800 m. Each node randomly chose a transmission range in $[r_{min}, r_{max}]$. For each network instance, we ran the test 1,000 times.

Fig. 5a compares the performance of three algorithms in terms of the CDS size. As shown in Fig. 5a, TSA is the best. In particular, the CDS size obtained from TSA is 11 percent smaller than that of WA and 4 percent smaller than that of TFA. Again, these results reveal that using the Steiner tree to interconnect a dominating set can reduce the CDS size.

As expected, $I_b < I_1 < I_s$, as shown in Fig. 5b. Note that I_s is 21 percent larger than I_b. This number is large and significant to increase the size of CDS. This very high percentage is predicted since, when k increases, K increases as well. Since $|I^*|/K \leq |I_b|$, $|I_b|$ has the potential to decrease as K increases.

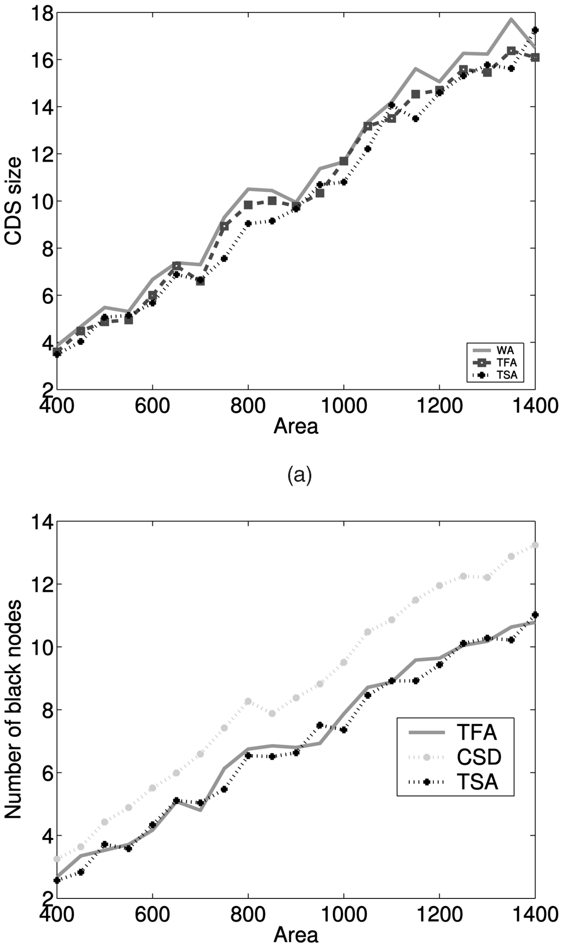
Fig. 5 illustrates how the transmission ranges affect the CDS size. As can be seen in Fig. 5, three curves show the obvious decreasing trend. In other words, the CDS size decreases when the maximum transmission range increases. It is due to the fact that, the larger the transmission range, the more nodes a node can dominate.

5.3 Effects of the Network Density

Simulations were also carried out to compare the performance of all three algorithms when changing the network density as well as to see how this change affects the CDS size. To change the network density, we fixed the number of nodes to $n = 50$ and increased the area from 400 m $\times$ 400 m to 1,400 m $\times$ 1,400 m with an increment of 50. In this experiment, we randomly generated 50 nodes in an area with the size changing as described. Each node randomly chose a transmission range in $[r_{min}, r_{max}]$, where $r_{min} = 200$ m and $r_{max} = 600$ m. For each network instance, we ran the simulations 1,000 times and averaged the results.

Fig. 6a provides the performance comparison of three algorithms in terms of the CDS size. As revealed in Fig. 6a, TSA still outperforms the other two in this case, and TFA outperforms WA. Specifically, the CDS size obtained from TSA is 8.2 percent less than that of WA and 3.2 percent less than that of TFA. As predicted, Fig. 6b indicates that $I_b < I_1 < I_s$. The number of black nodes obtained from TSA is slightly less than that of TFA but is much less than that of the CSD algorithm.

In addition, Fig. 6 shows the obvious increasing trend of three curves, which implies that the CDS size gets bigger when the network density decreases. This is because, when



(a)

(b)

Fig. 6. Effects of the network density. (a) Compare the CDS size. (b) Compare the MIS size.

the network density decreases, the number of neighbors of each node decreases as well. Thus, the CDS size needs to be larger to dominate all nodes in a network.

In conclusion, for all aspects that we have studied, TSA is the best algorithm. Next is TFA. Choosing nodes with the largest transmission ranges for the dominating set and using the Steiner tree with the minimum number of Steiner nodes to interconnect the dominating set can reduce the CDS size. Specifically, choosing nodes with the biggest radius can form a smaller dominating set. With the help of the Steiner tree, the number of blue nodes can be reduced. The size of a CDS obtained using these two mechanisms is about 10 percent less than that obtained without using them. In addition, the simulation results reveal that the CDS size increases as the number of nodes increases. The CDS size can also get larger if the network gets sparser. Furthermore, when the transmission ranges increase, the CDS size decreases.

6 Distributed Implementations

From the practical point of view, all algorithms designed in wireless networks should be distributed. In this section, we discuss how to implement our algorithms as distributed algorithms.

6.1 Distributed Version of TFA

There exist several distributed algorithms for constructing an MIS satisfying Lemma 2 in the literature [2], [3]. Specifically, Cidon and Mokryn constructed an arbitrarily rooted spanning tree T by the distributed leader-election algorithm in [16]. This algorithm has an $O(n)$ time complexity and $O(n \log n)$ message complexity, where n is the number of nodes in a given network. After constructing the spanning tree T, Wan et al. [2] introduced a distributed construction on how to find a maximal independent set using the color mechanism with $O(n)$ message complexity and $O(n)$ time complexity. We can use this construction for the first phase of our TFA and only need to present the distributed version of the second phase. Note that, after running the first phase, all nodes in an MIS are black and all other nodes are gray.

Algorithm 5 Distributed Version of TFA Second Phase

INPUT: An MIS I and $G = (V, E)$, all nodes $v_i \in I$ are black, and $v_j \in V - I$ are gray
OUTPUT: Color connectors blue
Set ID_C of each black node equal to the black nodes ID {ID_C is the black-blue component ID}
Set $ID_C = -1$ for all gray nodes v_j
Each gray node maintains the ADJ list which is the list of its adjacent black nodes in different black-blue components
Each gray node maintains a $COMPETITORS$ list
Each gray node maintains a global value B, $B = K$ initially
v_i sends a BLACK message contained its ID_C
Upon receiving the BLACK message, gray node v_j updates its ADJ and $COMPETITORS$ lists
v_j sends a GRAY message containing its id and its $|ADJ|$
v_j turns blue if its $|ADJ| > |ADJ|$ of its neighbors and its $|ADJ| \geq B$
Each blue node updates its ID_C to the smallest value in its ADJ list
A blue node then sends a BLUE message containing its new ID_C and new ADJ list
Upon receiving a BLUE message, black node v_i updates its ID_C and send a BLACK message
Upon receiving a BLUE message, a GRAY node decreases B by 1
If $|ADJ|$ of a gray node v_j equal to 1, then do nothing

As described in Algorithm 5, each black node v_i in the MIS I maintains its black-blue component ID, i.e., ID_C. Initially, we have $|I|$ black-blue components. Hence, the ID_C of each black node can be set to the node ID. Each gray node maintains its black-blue component id and, initially, $ID_C = -1$, which indicates that it does not belong to any black-blue component yet. Each gray node also maintains a list of its adjacent black-blue components with their ID_C values, called ADJ, and a list of its competitors, called $COMPETITORS$. The gray node is adjacent to a black-blue component if it is adjacent to a *black* node in the black-blue component. A gray node u is a competitor of a gray node v if the number of adjacent black-blue components of v and u are the same. For breaking ties between competitors, the node with the smaller node id becomes a blue node. Hence, the $COMPETITORS$ list contains a list of competitor nodes IDs. Each gray node also maintains a global value B, which represents the maximum number of independent neighbors. Initially, $B = K$.

Note that, after finding a maximal independent set, we still have a spanning tree T. Thus, each node also maintains a list of its children in T, called $CHILDREN$. Initially, a root node of T which is a black node sends a BLACK message containing its ID_C to its one-hop neighbors. Upon receiving a BLACK message, the gray node v_j adds the ID_C in the BLACK message to its adjacent black-blue components ADJ. If this number ID_C is already in ADJ, it does nothing. After updating its ADJ, the gray node then broadcasts a GRAY message $< |ADJ|, id >$. Note that id is the gray node ID and $|ADJ|$ is the size of the ADJ list. Upon receiving the GRAY message, a gray node compares its $|ADJ|$ to the $|ADJ|$ in the GRAY message. If its $|ADJ|$ is equal to the $|ADJ|$ in the GRAY message, it adds the gray node id in the GRAY message to its $COMPETITORS$ list.

When a node is a leaf, in addition to broadcasting the BLACK or GRAY message based on its color, it also broadcasts an END message. Upon receiving the END message, a GRAY node turns blue if the following conditions are satisfied:

- its $|ADJ| \geq B > 1$ and
- its id is smaller than all ids in its $COMPETITORS$ list.

After turning its color to blue, a blue node updates its ID_C to the smallest number in its ADJ list and decreases B by 1. The blue node then sends a BLUE message and keeps its color permanent. The BLUE message contains its id and its ADJ list. Upon receiving a BLUE message, all black nodes update their ID_C to the smallest number in the BLUE message and send the BLACK message out. Note that all nodes in the same black-blue component must have the same ID_C. At the end of this algorithm, a gray node keeps its color gray if its $|ADJ|$ is 1. A node stops sending a message if it is adjacent to one black-blue component. This indicates that either all black nodes are connected at this time or a node is just adjacent to only one black node. The main idea of this distributed version is shown in Algorithm 5.

Theorem 3. *The distributed version of TFA has an $O(n \log n)$ message complexity and $O(n)$ time complexity.*

Proof. The time and message complexity of the MIS construction phase is dominated by the time and message complexity of constructing the rooted spanning tree T, which are $O(n)$ and $O(n \log n)$, respectively [2]. For the second phase, each node sends at most $O(n \log n)$ messages and takes at most linear time. Hence, the message complexity of distributed TFA is $O(n \log n)$, where its time complexity is $O(n)$. □

6.2 Distributed Version of TSA

The distributed version of TSA consists of two phases, as shown in Algorithm 6. The first phase is to find a dominating set such that, at each iteration, we select a node with the largest transmission range. The second phase is to connect the above dominating set.

Algorithm 6 Distributed Version of TSA

INPUT: A DGB $G = (V, E)$ with all nodes in white
OUTPUT: A CDS
Each white node maintains a $SORT$ list
Each white node v_i broadcasts a WHITE message $< id_i, r_i >$
Upon receiving a WHITE message, each node updates its $SORT$ list
A node with its id at the beginning of the $SORT$ list marks itself black and sends the BLACK message contained its id
Upon receiving a BLACK message, a white node marks itself gray and broadcasts the GRAY message contained its id and id in the BLACK message
Upon receiving a GRAY message, a white node updates its $SORT$ list
Use the Algorithm 5 to connect all black nodes

Initially, all nodes are white. Each node maintains a list of all node ids in the decreasing order of the transmission ranges, called $SORT$ list. At the beginning, the $SORT$ list of each node contains its own id. Each white node v_i broadcasts a WHITE message containing its own id and its transmission range $< id_i, r_i >$. Upon receiving a WHITE message, each node updates its $SORT$ list by adding the id to the WHITE message in the decreasing order of transmission ranges. A node which has its own id at the head of the $SORT$ list has the largest transmission range.

A white node which has its own id at the head of the $SORT$ list marks itself black and sends a BLACK message to its neighbors. The BLACK message contains the black node id. Upon receiving a BLACK message, a white node marks itself gray. The gray node then broadcasts a GRAY message which contains its own id and id in the BLACK message. Upon receiving the GRAY message, a white node updates its $SORT$ list by removing the id in the gray message from the $SORT$ list.

Once a node marks itself black or gray, its color remains unchanged. This process stops when there does not exist any white node. Note that, after marking itself black or gray and sending out the BLACK or GRAY message, this node will not participate in the coloring process anymore.

At the end of this phase, all nodes in the network are either black or gray. All black nodes form a dominating set. Now, we need to connect these black nodes. The process is similar to the distributed version of TFA Second Phase.

Theorem 4. *The distributed version of TSA has an $O(n^2)$ message complexity and $O(n^2)$ time complexity.*

Proof. The time and message complexity of the first phase is dominated by the sorting part, i.e., to compute the $SORT$ list of each node. Since each node broadcasts a WHITE message, the time and message complexity is $O(n^2)$. The second phase uses an $O(n \log n)$ message and takes at most linear time. Hence, the message complexity of distributed TSA is $O(n^2)$ and its time complexity is also $O(n^2)$. □

7 Conclusions

In this paper, we have studied the Connected Dominating Set (CDS) problem in Disk Graphs with only Bidirectional links (DGB). Disk graphs can be used to model wireless ad hoc networks where nodes have different transmission ranges. We have proposed two approximation algorithms and shown that the size of the obtained CDS is within a constant factor of that of the optimal CDS. The main approach in our algorithms is to construct a maximal independent set and then connect them. Through theoretical analysis and simulation experiments, we have shown that using a Steiner tree with the minimum number of Steiner nodes to interconnect the maximal independent set can help to reduce the size of the CDS. In addition, choosing a node with the largest transmission range as a dominator can further reduce the CDS size.

Moreover, we have proven the size relationship between an independent set and a CDS. We have also pointed out some important properties of a DGB. In particular, given a DGB G, there exists a node such that the maximum number of its independent neighbors is 5. In addition, we have shown the upper bound of the maximum number of independent neighbors of any node in a DGB.

When nodes in a network have different transmission ranges, a node u may be able to communicate directly to a node v but node v may not be able to respond directly back to node u. In this case, the edge (u, v) is a directed edge, called a unidirectional link. Thus, we are interested in studying the CDS problem in directed disk graphs, where both unidirectional and bidirectional links may exist. One viable solution is to find a dominating set and then use a directed Steiner nodes algorithm to connect them.

References

[1] B. Clark, C. Colbourn, and D. Johnson, "Unit Disk Graphs," *Discrete Math.*, vol. 86, pp. 165-177, 1990.

[2] P.-J. Wan, K.M. Alzoubi, and O. Frieder, "Distributed Construction on Connected Dominating Set in Wireless Ad Hoc Networks," *Proc. INFOCOM*, 2002.

[3] M. Cardei, M.X. Cheng, X. Cheng, and D.-Z. Du, "Connected Domination in Ad Hoc Wireless Networks," *Proc. Sixth Int'l Conf. Computer Science and Informatics (CSI '02)*, 2002.

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

[5] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and L.-I. Ko, "A Greedy Approximation for Minimum Connected Dominating Sets," *Theoretical Computer Science*, vol. 329, nos. 1-3, pp. 325-330, 2005.

[6] J. Wu and H. Li, "On Calculating Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm.*, pp. 7-14, Aug. 1999.

[7] Y. Li, S. Zhu, M.T. Thai, and D.-Z. Du, "Localized Construction of Connected Dominating Set in Wireless Networks," *Proc. US Nat'l Science Foundation Int'l Workshop Thoretical Aspects of Wireless Ad Hoc, Sensor and Peer-to-Peer Networks*, 2004.

[8] Y. Li, M.T. Thai, F. Wang, C.-W. Yi, P.-J. Wang, and D.-Z. Du, "On Greedy Construction of Connected Dominating Sets in Wireless Networks," *Wireless Comm. and Mobile Computing*, vol. 5, no. 8, pp. 927-932, 2005.

[9] K.M. Alzoubi, P.-J. Wang, and O. Frieder, "Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks," *Proc. MobiHoc*, 2002.

[10] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in Ad Hoc Networks Using a Spine," *Proc. Int'l Conf. Computers and Comm. Networks*, 1997.

[11] B. Das and V. Bharghavan, "Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets," *Proc. Int'l Conf. Comm.*, 1997.

[12] R. Sivakumar, B. Das, and V. Bharghavan, "An Improved Spine-Based Infrastructure for Routing in Ad Hoc Networks," *Proc. IEEE Symp. Computers and Comm.*, 1998.

[13] M.R. Garey and D.S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[14] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks," *Proc. 35th Hawaii Int'l Conf. System Sciences*, 2002.

[15] K.M. Alzoubi, P.-J. Wan, and O. Frieder, "Distributed Heuristics for Connected Dominating Sets in Wireless Ad Hoc Networks," *J. Comm. and Networks*, vol. 4, no. 1, Mar. 2002.

[16] I. Cidon and O. Mokryn, "Propagation and Leader Election in Multihop Broadcast Environment," *Proc. 12th Int'l Symp. Distributed Computing (DISC '98)*, 1998.

[17] J. Wu, "Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links," *IEEE Trans. Parallel and Distributed Computing*, vol. 13, no. 9, pp. 866-881, Sept. 2002.

[18] F. Dai and J. Wu, "An Extended Localized Algorithms for Connected Dominating Set Formation in Ad Hoc Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 15, no. 10, Oct. 2004.

[19] D.-Z. Du, L. Wang, and B. Xu, "The Euclidean Bottleneck Steiner Tree and Steiner Tree with Minimum Number of Steiner Points," *Proc. Ann. Int'l Computing and Combinatorics Conf. (COCOON '01)*, Aug. 2001.

[20] D. Chen, D.-Z. Du, G.-H. Lin, L. Wang, and G. Xue, "Approximations for Steiner Trees with Minimum Number of Steiner Points," *J. Global Optimzation*, vol. 18, pp. 17-33, 2000.

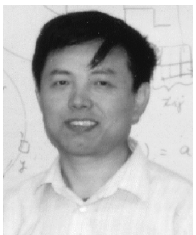
[21] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli, "Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison," *IEEE Trans. Parallel and Distributed Computing*, vol. 17, no. 4, pp. 292-306, Apr. 2006.

My T. Thai received the BS degree in computer science and the BS degree in mathematics from Iowa State University in 1999. She received the PhD degree from the University of Minnesota, Twin Cities, in 2006. She is now an assistant professor in the Department of Computer and Information Sciences and Engineering at the University of Florida. Her current research interests include combinatorics, algorithms, wireless networks, and computational biological. She is a member of the IEEE.

Feng Wang received the PhD degree in computer science from the University of Minnesota in 2005. Her research interests include wireless networks, computational biology, and combinatorial optimization. She is currently working at Seagate Technology. She is a member of the IEEE.

Dan Liu is a PhD candidate in the Computer Science and Engineering Department at the University of Minnesota. Her research interests include wireless networks and algorithms. She is a student member of the IEEE.

Shiwei Zhu is a PhD candidate in the Computer Science and Engineering Department at the University of Minnesota. His research interests include wireless networks and optimization. He is a student member of the IEEE.



Ding-Zhu Du received the MS degree from the Chinese Academy of Sciences in 1982 and the PhD degree from the University of California at Santa Barbara under the supervision of Professor Ronald V. Book in 1985. Before settling at the University of Texas at Dallas, he worked as a professor in the Department of Computer Science and Engineering at the University of Minnesota. He also worked at the Mathematical Sciences Research Institute in Berkeley for one year, in the Department of Mathematics at the Massachusetts Institute of Technology for one year, and in the Department of Computer Science at Princeton University for one and a half years. Currently, he has published more than 140 journal papers and several books. He is the editor-in-chief of the *Journal of Combinatorial Optimization* and is also on the editorial boards for several other journals. Thirty PhD students have graduated under his supervision. He is a member of the IEEE.