Distributed Connected Dominating Set Construction in Geometric k-Disk Graphs

Kai Xing & Wei Cheng
Computer Science
The George Washington University
Washington, DC 20052, USA
{kaix, wcheng}@gwu.edu

E. K. Park [*]
Computer Science & Electrical Engineering
University of Missouri at Kansas City/NSF
Kansas City, MO 64110, USA
ekpark@umkc.edu

Shmuel Rotenstreich
Computer Science
The George Washington University
Washington, DC 20052, USA
shmuel@gwu.edu

Abstract

In this paper, we study the problem of minimum connected dominating set in geometric k-disk graphs. This research is motivated by the problem of virtual backbone construction in wireless ad hoc and sensor networks, where the coverage area of nodes are disks with different radii. We derive the size relationship of any maximal independent set and the minimum connected dominating set in geometric k-disk graphs, and apply it to analyze the performances of two distributed connected dominating set algorithms we propose in this paper. These algorithms have a bounded performance ratio and low communication overhead, and therefore have the potential to be applied in real ad hoc and sensor networks.

1 Introduction

Let V be the set of points denoting the set of nodes randomly placed in the Euclidean plane. Associated with $\forall u \in V$ is a radius r_u such that $1 \le r_u \le k$, where k is a constant. Let $G_k(V, E)$ be the disk graph constructed from V such that an edge $(u, v) \in E$ if and only if the distance between u and v is at most $\min\{r_u, r_v\}$. We call G_k a *geometric k-disk graph* or a *geometric k-disk graph*. When $k = 1$, G_k is a unit-disk graph.

Geometric k-disk graphs have been widely adopted to model wireless ad hoc and sensor networks, in which two nodes can communicate with each other successfully via a two-way handshake (DATA-ACK) if they reside in each other's transmission range. Due to different power capacities or for the reason of interference mitigation, nodes may have different transmission ranges, resulting in disks with different radii.

In this paper we study the problem of constructing minimum connected dominating sets (MCDS) in geometric k-disk graphs. One significant application of this problem is the virtual backbone construction in ad hoc and sensor networks, where a virtue backbone is utilized to decrease the protocol overhead [4,10]. Computing a MCDS in $G_k(V, E)$ is a NP-hard problem since its special case, MCDS in unit-disk graphs, is NP-hard [7]. In this paper, we first figure out the size relationship between any maximal independent set (MIS) and the minimum connected dominating set in G_k. Then we report two distributed approximation algorithms for connected dominating set construction.

Our major contributions are three folds. First, we prove that the maximum number of independent neighbors a node may have in a geometric k-disk graph is at most $5 + 9\lceil \frac{\ln k}{\ln(2\cos(\frac{\pi}{5}))} \rceil$. Second, we derive the size relationship between any MIS and the MCDS in geometric k-disk graphs, which is $(8\frac{2}{3}\lceil \frac{\ln k}{\ln(2\cos(\frac{\pi}{5}))} \rceil + 4\frac{5}{6})opt$, where opt is the size of the MCDS. Note that the best results in the literature were reported by Thai et al. in [14], which stated that when $k > 1$, each node has at most $10 + 10\lceil \frac{\ln k}{\ln(2\cos\frac{\pi}{5})} \rceil$ number of independent neighbors and the size of any MIS

[*] This material was based on work supported by the National Science Foundation, while working at the Foundation.

is at most $(10 + 10\lceil \frac{\ln k}{\ln(2\cos\frac{\pi}{5})} \rceil)opt$. Notice that our results work for unit-disk graphs too, where $k = 1$.

The third contribution of this paper is the two proposed distributed approximation algorithms, for which we conduct rigorous theoretical performance analysis. In the first algorithm, we grow a CDS from a leader. This algorithm requires each node know the ids of its neighbors. In the second algorithm, we construct an MIS first then find out connectors to connect all nodes in the MIS. This algorithm takes node degree into consideration and therefore has a higher message overhead but achieves a better performance compared to the first one.

This paper is organized as follows. We first introduce the preliminary definitions in Section 2. Related works are also surveyed in this section. Then we derive the size relationship between any maximal independent set and the minimum connected dominating set in geometric k-disk graphs in Section 3. The two distributed approximation algorithms are reported in sections 4 and 5, respectively. We conclude this paper with a brief summary and a brief discussion on future research in Section 6.

2 Preliminaries And Related Work

2.1 Preliminaries

Given any graph $G = (V, E)$, two vertices are *independent* if they are not neighbors. For $\forall\ u, v \in V$, $hop\_count(u, v)$ is the number of edges (hops) in the shortest path from u to v. For any vertex v, $N_1[v] = \{u \mid hop\_count(u, v) \leq 1\}$ is the *one-hop (close) neighbor set* of v; $N_2[v] = \{u \mid hop\_count(u, v) \leq 2\}$ is the *two-hop (close) neighbor set* of v.

An *independent neighbor set* of v, denoted by $N_I(v)$, is a subset of $N_1[v]$ such that any pair of vertices in $N_I(v)$ are independent. An *independent set* S of G is a subset of V such that for $\forall\ u, v \in S$, $(u, v) \notin E$. S is *maximal* if any vertex not in S has a neighbor in S.

A *dominating set* D of G is a subset of V such that each node not in D has at least one neighbor in D. For any edge (u, v), if $u \in D$ and $v \notin D$, then u is v's *dominator* and v is u's *dominatee*. If both u and v are $\in D$, one can specify the other as its dominator. An optimal dominating set has minimum cardinality. If the induced subgraph of D is connected, then D is a *connected dominating set (CDS)*. Among all CDSs of graph G, the one with minimum cardinality is called a *minimum connected dominating set (MCDS)*. A maximal independent set is also a dominating set.

2.2 Related Work

To our best knowledge, Thai et al. [14] was the first and the only one to tackle the problem of MCDS construction in geometric k-disk graphs. In that work, the authors proved that any node can have at most $10+10\lceil \frac{\ln k}{\ln(2\cos\frac{\pi}{5})} \rceil$ independent neighbors when $k > 1$. Thai et al. [14] also proposed three centralized approximation algorithms together with their performance analysis. Our work is motivated by [14] but improves [14]. In addition, we focus on the design of distributed algorithms for the geometric k-disk graphs.

There exist abundant works investigating the problem of MCDS construction in unit-disk graphs and general graphs. For a recent literature survey, we refer the readers to [10] and the references therein. In the following we briefly summarize several major works.

The NP-Completeness of MCDS in general graphs was studied in [8]. The MCDS remains NP-hard for unit-disk graphs [7]. In 1998, Guha and Khuller proposed two CDS construction strategies in their seminar work [9]. These two greedy heuristic algorithms have performance ratios $2(1 + H(\Delta))$ and $3+\ln\Delta$, respectively, where Δ the largest node degree and H is the harmonic function. Ruan *et al.* [13] proposed a one-step greedy approximation algorithm with a performance ratio of $2 + \ln\Delta$. Wu and Li [16] proposed the first localized algorithm for MCDS in general graphs but their algorithm does not have a bounded performance guarantee [15].

For unit-disk graphs, Wu *et al.* [17] studied the size relationship between MCDS and MIS and reported that the size of any MIS is at most 3.8 times of that of MCDS plus a constant 1.2. With this result, the performance ratio for the approximation algorithms in [1,3,15], which connect a MIS with a spanning tree, is improved from 8 to 7.8. Li *et al.* [12] designed the algorithm of connecting an MIS with a Steiner tree, and achieved a performance ratio of $5.8 + \ln 4$. Note that all the above-mentioned works require a MIS satisfying the follwing property: any subset of the MIS is two-hop away from its complementary. The construction of such a MIS relies on a spanning tree. The works by Wan *et al.* [2] and Cheng *et al.* [4] removed this restriction with a tradeoff of a larger performance ratio. MCDS in unit-disk graphs has a polynomial time approximation scheme [5], which means that MCDS in unit-disk graphs can be approximated to any degree.

The research reported in this paper is motivated by [4] and [14] and we target the geometric k-disk graphs. To our knowledge, the results reported in this paper are the best in the literature.

3 Maximal Independent Set and Minimum Connected Dominating Set in Geometric k-disk Graphs

Given $G_k(V, E)$, a geometric k-disk graph as defined in Sec 1, let $N_I(x)$ be the set of independent neighbors of x for $\forall x \in V$. We have

Lemma 3.1. $|N_I(x)| \leq 5 + 9\lceil \frac{\ln k}{\ln(2\cos(\frac{\pi}{5}))} \rceil$.

Proof. Our proof is motivated by [14]. Denote by r_x the radius of x. Let α be a real number such that $0 < \alpha < \frac{\pi}{3}$. For easier elaboration we assume that $\frac{2\pi}{\alpha}$ is a constant integer and we set $\alpha = \frac{\pi}{5}$. We will explain at the end of the proof that this α value is a reasonable choice. Note that the entire proof procedure only requires $0 < \alpha < \frac{\pi}{3}$.

We first draw circles centered at x with radii $(2\cos\alpha)^0$, $2\cos\alpha$, $(2\cos\alpha)^2$, $\cdots$, $(2\cos\alpha)^{n_s-1}$, and $(2\cos\alpha)^{n_s}$, respectively, such that $(2\cos\alpha)^{n_s-1} < r_x$ and $(2\cos\alpha)^{n_s} \geq r_x$. Therefore $n_s = \lceil \frac{\ln r_x}{\ln(2\cos\alpha)} \rceil$. By this way the disk of x is partitioned into n_s annuluses plus the unit-radius circle centered at x, as shown in Fig. 1. We claim that each annulus contains at most $\frac{2\pi}{\alpha} - 1$ nodes in $N_I(x)$.

Consider any annulus $\mathcal{H}$, as shown in Fig. 1(a). Let u and v be two independent neighbors of x in $\mathcal{H}$. We must have $\angle uxv > \alpha$. To prove this claim, we draw two lines xb and xd crossing the inner and outer sides of $\mathcal{H}$ at a and b, and c and d, respectively, such that u resides in xb and $\angle bxd = \alpha$. Since $\frac{|xb|}{|xc|} = \frac{|xd|}{|xa|} = 2\cos\alpha$, we have $|cb| = |ad| = |xa| = |xc|$. Therefore $|yx| > |yb|$ holds true for any point y in the closed area $abdc$ of $\mathcal{H}$, which means all nodes in the closed area of $abdc$ in $\mathcal{H}$ are neighbors of u. Thus v must be out of the area of $abdc$, and therefore $\angle uxv > \alpha$. Based on this argument, we conclude that there exist at most $\frac{2\pi}{\alpha} - 1$ independent neighbors of x in the annulus $\mathcal{H}$.

Since there are n_s annuluses, the total number of independent neighbors of x in all the annuluses is at most $n_s \cdot [\frac{2\pi}{\alpha} - 1] = \lceil \frac{\ln r_x}{\ln(2\cos\alpha)} \rceil \cdot [\frac{2\pi}{\alpha} - 1]$. Let $f(\alpha) = \lceil \frac{\ln r_x}{\ln(2\cos\alpha)} \rceil \cdot [\frac{2\pi}{\alpha} - 1]$. When $0 < \alpha < \frac{\pi}{3}$, $f(\alpha)$ yields a minimum when α is close to $\frac{\pi}{5}$.

Now consider the unit disk $\mathcal{D}$ centered at x, as shown in Fig. 1(b). Let u and v be two independent neighbors of x. Then we must have $\angle uxv > 60°$, since otherwise, $|uv| \leq \max\{|ux|, |vx|\} \leq 1$, contradicting the independence of u and v. Therefore there exist at most 5 independent neighbors of x at $\mathcal{D}$.

Taking into account the unit-disk area and all the annulus areas, we have $|N_I(x)| \leq 5 + 9\lceil \frac{\ln r_x}{\ln(2\cos(\frac{\pi}{5}))} \rceil \leq 5 + 9\lceil \frac{\ln k}{\ln(2\cos(\frac{\pi}{5}))} \rceil$. □

In the following analysis, we denote $\lceil \frac{\ln k}{\ln(2\cos(\frac{\pi}{5}))} \rceil$ by n_k. Now we have derived the upper bound of the number of independent neighbors for any node x in $G_k(V, E)$, which is $5 + 9n_k$. Next, we are going to answer the following question: how many independent neighbors a pair of neighboring nodes may have? The following lemma answers this question by giving an upper bound.

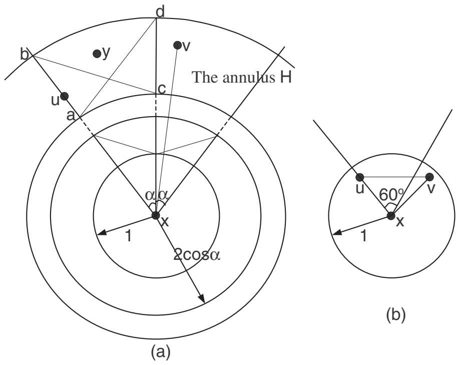

Figure 1. (a) If u and v are independent neighbors in the annulus $\mathcal{H}$, then $\angle uxv > \alpha$. Note that only part of $\mathcal{H}$ is shown. (b) When u and v are independent neighbors in the unit-disk area of x, $\angle uxv > 60°$.

Lemma 3.2. *Let u and v be any pair of neighboring nodes in $G_k(V, E)$, i.e. $u \in V$, $v \in V$, and $(u, v) \in E$. Then the total number of independent neighbors of u and v is at most $9 + 16n_k$.*

Proof. Let r_u and r_v be the radii of u and v, respectively. Without loss of generality, we assume $r_u \geq r_v$. Assume the disks of u and v intersect at x and y, as illustrated in Fig. 2. We have $|ux| = r_u$ and $|vx| = r_v$. Since $|ux| \geq |vx|$ and $|ux| \geq |uv|$, we have $\angle uvx \geq \frac{\pi}{3}$. Similarly we have $\angle uvy \geq \frac{\pi}{3}$. Therefore $\angle xvy \geq \frac{2\pi}{3}$. Thus the area in v's disk that could cover nodes in $N_I(v)$ but not in $N_I(u)$ is the right sector $\widehat{xvy}$ of v with an angle of at most $\frac{4\pi}{3}$.

Now we can take the same procedure as that in Lemma 3.1 by drawing circles centered at v with radii 1, $2\cos\alpha, \cdots$. Since the angle at v formed by any two independent neighbors of v in the same annulus is larger than α, the overlapping area of the right sector $\widehat{xvy}$ and all the annuluses can contain at most $(\lceil \frac{\frac{4\pi}{3}}{\alpha} \rceil) n_k = 7n_k$ independent neighbors of v. With a similar argument, the intersection of the unit-disk centered at v and the right sector $\widehat{xvy}$ may contain at most $\frac{\frac{4\pi}{3}}{\frac{\pi}{3}} = 4$ independent neighbors. Therefore the total number of independent neighbors of u and v is at most $9 + 16n_k$. □

Lemma 3.2 indicates that a neighbor can bring in at most $4 + 7n_k$ independent nodes. This motivates us to prove the following lemma addressing the size relationship between a

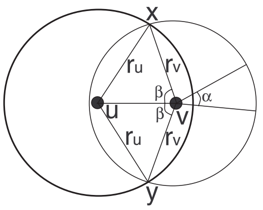

Figure 2. If u and v are neighbors in $G_k(V, E)$ and $r_u \geq r_v$, then $\angle xvy > \frac{2\pi}{3}$.

minimum connecting dominating set and any maximal independent set of the same G_k.

Let $S \subset V$ be any MIS of $G_k(V, E)$. Denote by opt the size of any MCDS in G. We have

Theorem 3.1. $|S| \leq (4\frac{5}{6} + 8\frac{2}{3}n_k)opt.$

Proof. Let C be any MCDS in $G_k(V, E)$, and opt denote its size. Compute a spanning tree τ traversing all nodes in C. Let $u_1, u_2, \cdots, u_{opt}$ be the nodes decreasingly ordered by radius in τ.

Given an arbitrary node u_i in τ, letting $size(u_i)$ be the maximum number of independent neighbors it dominates in S that are not dominated by its neighbors, we have

- Case 1: $size(u_i) \leq 5 + 9n_k$, if $i < j$ for any of its neighbors u_j in τ (namely u_i has the lowest index (largest radius) among its neighbors in τ);
- Case 2: $size(u_i) \leq 4 + 7n_k$, otherwise.

In case 1, according to Lemma 3.1, u_i will dominate at most $5 + 9n_k$ independent nodes in S. In case 2, let u_j be one of the nodes with lower index (larger radius) than that of u_i. Based on Lemma 3.2, since $j < i$, u_i will dominate at most $4 + 7n_k$ independent nodes in S that are not dominated by u_j.

Let t denote the number of nodes belonging to case 1. We are going to prove $\frac{t}{opt-t} \leq \frac{5}{1}$. Given an arbitrary node u in τ, each of its neighbors belonging to case 1 will dominate at least $\frac{2\pi}{3}$ sector of u's disk. Therefore, u has at most $\lfloor \frac{2\pi}{\pi/3+\varepsilon} \rfloor = 5$ neighbors belonging to case 1 (otherwise they are within each other's disk and contradict the condition of case 1).

Since C is a connected dominating set, each node in S will be either in C, or dominated by one node in C. Therefore $|S| \leq (opt - t)(4 + 7n_k) + t(5 + 9n_k) \leq (4\frac{5}{6} + 8\frac{2}{3}n_k)opt.$ □

In the following sections, we propose two distributed CDS algorithms and analyze their performance theoretically. Note that for both algorithms, we assume that each node has a unique id, known to its neighbors within a two hops distance.

4 Algorithm I: Growing A CDS from A Leader

Algorithm I starts from a leader node. For simplicity, we assume that the node with the smallest id is the leader. We associate a color with each node. Strictly speaking, *color* is not a parameter in our algorithm. It is retained in the algorithm description for the purpose of easier elaboration.

Initially all nodes are colored white. In the first step, leader u colors itself black and becomes a dominator. Then all nodes in $N_1[u] \setminus \{u\}$ color themselves gray and become dominatees, and all nodes in $N_2[u] \setminus N_1[u]$ color themselves yellow and become active. In the next step, neighboring active nodes compete with each other and the winners, whose ids are the smallest among their yellow neighbors, become dominators. Each of the winners also specifies its own dominator, the gray neighbor whose id is the smallest among all gray neighbors. This step will repeat until nodes are either gray (dominatees) or black (dominators). All black nodes form a CDS.

Note that the procedure described above grows a CDS from the leader. In the first step, only the leader is included in the CDS. In each of the other steps, two nodes are included in the CDS, with one turning color from yellow and one from grey.

Theorem 4.1. *The size of the CDS generated by Algorithm I is at most* $(9\frac{2}{3} + 17\frac{1}{3}n_k)opt - 1$*, where opt is the size of the minimum connected dominating set.*

Proof. We can partition all the dominators into two sets: A and B. Set A contains all vertices with color changing from *white to black directly* and B contains all vertices with color changing from *white to gray then to black.* The first step adds the leader to A. Each of the other steps adds one node to A and one node to B. Thus $|A| = |B| + 1$.

Now we claim that A is an independent set. This is obvious since each vertex u in A is colored black from white or yellow. This means u has no black neighbors because each neighbor of a black node has gray color. From Theorem 3.1, $|A| \leq (4\frac{5}{6} + 8\frac{2}{3}n_k)opt.$ Thus $|A| + |B| \leq (9\frac{2}{3} + 17\frac{1}{3}n_k)opt - 1.$ ■

Remarks: (i) Algorithm I grows a tree from the leader in a step-by-step fashion. At any time, all the inner nodes of the tree are colored black while all the leaf nodes are colored gray. In the first step, the leader and all of its neighbors are added to the tree. In every other step, a leaf node v and one of its yellow neighbors u are colored black (added to the tree). All the white/yellow neighbors of u and v are colored gray and added to the tree as leaves. This algorithm terminates when no white/yellow node is left. All the black nodes form the CDS. (ii) If we use node cost instead of id as the criteria for dominator selection, Algorithm I is cost-aware. For example, if the cost is the inverse of the residual power of each node, the output CDS has a higher power capacity; if the cost is the incoming bitrate (load), the output CDS has a lower load; if the cost is the inverse of the node velocity, the induced graph by the output CDS has a more stable topology.

4.1 Distributed Implementation

Each node u maintains the following parameters: dom_u, which is the dominator, or the parent of the node in the tree; $rank_u$, which defines a relative relationship among neighboring nodes[1]; $children_u$, which contains all dominated nodes, or the children of the node in the tree. These parameters are updated by the exchange of the following 3 messages.

$< dominator(u, dom_u, rank_u) >$ — node u, whose dominator is dom_u and whose rank is $rank_u$, broadcasts this message to all neighbors. u is a dominator.

$< dominatee(u, dom_u, rank_u) >$ — node u, whose dominator is dom_u and whose rank is $rank_u$, broadcasts this message to all neighbors. u is a dominatee.

$< active(u) >$ — node u broadcasts this message to all of its white/yellow neighbors when it becomes active. A white node becomes active after it receives the first $<$ $dominatee >$ message from one of its neighbors.

The state transition diagram of Algorithm I is given in Figure 3. Each node u runs a copy of the algorithm. At any time, u can be in one of the 4 states: S_0, S_1, S_2, and S_3. The directed arc from S_i to S_j, where $i, j = 0, 1, 2, 3$, represents the transition from state S_i to state S_j. Each transition is labeled by a number. These transitions will be explained latter.

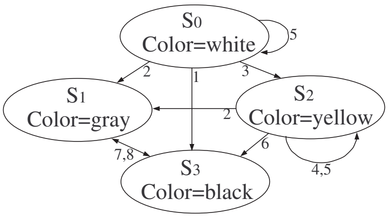

Figure 3. *The state transition diagram of Algorithm I for any node u.*

State S_0 is the initial state. A node in this state has white color. All nodes are in S_0 at the beginning of the algorithm. State S_1 is the dominatee state. A node in this state is a dominatee and has gray color. State S_2 is the active state. A node in this state has at least one neighbor in S_1 and has yellow color. An active node is a candidate dominator in next step. State S_3 is the dominator state. A node in this state is a dominator and has black color. All nodes in S_3 form the connected dominating set. Each node u in state S_0 or S_2 also maintains a parameter W_u including its all yellow neighbors. Initially $W_u = \emptyset$. If u has the smallest id compared to its yellow neighbors in W_u, u will become a dominator in the next step. The transition steps are detailed below.

1. u is in state S_0. If u is the leader, then $dom_u = u$, $rank_u = 0$. u will broadcast message $< dominator(u, u, 0) >$ and go to state S_3.

2. u is in state S_0 or S_2 and receives message $< dominator(v, d, l) >$ from neighbor v. If d is a neighbor of u, then $dom_u = d$, $rank_u = l$; otherwise, $dom_u = v$, $rank_u = l + 1$. u will broadcast message $< dominatee(u, dom_u, rank_u) >$ and go to S_1.

3. u is in state S_0 and receives message $< dominatee(v, d, l) >$ from neighbor v. u will broadcast message $< active(u) >$ and go to state S_2. u temporarily sets dom_u to v and $rank_u$ to $l + 1$. If v is in W_u, it will be removed.

4. u is in state S_2 and receives message $< dominatee(v, d, l) >$. Remove v from W_u if it is in W_u. If $id_v < id_{dom_u}$, then $dom_u = v$ and $rank_u = l + 1$. u will go back to S_2.

5. u is in state S_0 or S_2 and receives message $< active(v) >$. $W_u = W_u \cup \{v\}$. u remains in the original state.

6. u is in state S_2 and there is no broadcasting in $N_1[u]$ for T_0 time unit (which is a design constant). If u has the smallest id compared with all nodes in W_u, it will broadcast message $< dominator(u, dom_u, rank_u) >$ and go to S_3.

7. u is in state S_1 and receives message $< dominator(v, d, l) >$ from v. If $d = u$, u will broadcast message $< dominator(u, dom_u, rank_u) >$ and go to state S_3.

8. u is in state S_3. If u finds that none of its neighbors takes u as the dominator, u will broadcast message $dominatee(u, dom_u, rank_u)$ and go to state S_1.

Remarks: (i) The leader can be elected by the distributed leader election algorithm proposed in [6]. (ii) The parameter T_0 is used to force the start of next step in nearby environment be strictly after the end of the current step. T_0 is

[1] In Algorithm I, rank is the level of the node in the tree

a design parameter, which could be predetermined via simulation. (iii) We assume the message broadcast is reliable. For reliable broadcasting, we refer the readers to [11](iv) When u receives a message $< \mathit{dominator}(v, u, l) >$ or $< \mathit{dominatee}(v, u, l) >$, u will put v into its children list if v is not there. (v) In state S_1, a gray node will select the black node with the lowest rank as its dominator. This can help to optimize the generated CDS. For example, if a black node u's dominated children are also dominated by $\mathit{dom}(u)$, then u can become a dominatee and go to state S_1. (vi) Step 8 is an optimization to remove those dominators with no dominatees.

Theorem 4.2. *Algorithm I has a message complexity $O(n)$, where n is the total number of vertices.*

Proof. We have 3 types of messages: $< \mathit{dominator} >$, $< \mathit{dominatee} >$, and $< \mathit{active} >$. Each node broadcasts each message at most once. Thus the total number of broadcastings is at most $O(n)$. ■

Note that the above message complexity analysis does not consider the leader election, whose message complexity is $\Omega(n \log n)$ [2].

5 Algorithm II: Connecting A Maximal Independent Set

Algorithm I could be improved if node degree is taken into consideration. In Algorithm II, instead of growing a CDS from a leader, we first compute a MIS, then connect all nodes in the MIS. Accordingly Algorithm II contains two phases.

In Algorithm II, each node needs to maintain two more parameters: the *effective degree* d^*, which is defined to be the number of white/yellow neighbors, and the *black degree* d_b^*, which is used by a gray node to record the number of black neighbors with a higher rank. The distributed implementation of algorithm II is detailed in the following subsection.

5.1 Distributed Implementation

The state transition diagram for the first phase is shown in Figure 4.

Initially all nodes are colored white. During the execution of phase 1, each white/yellow node keeps track of its effective degree d^*, which will be updated when a neighbor changes color. To keep the degree information up-to-date, we need a new message $< \mathit{degree}(u, d^*) >$, by which u tells its neighbors that its effective degree is d^*. We also need the message $< \mathit{blackdegree}(u, d_b^*) >$, which is broadcasted by u, a dominatee. The "rank" information will be used in phase 2 to connect all members in the generated MIS. The transitions are explained below. Initially $d^* = d$, where d is the degree whose value is $|N_1[u]| - 1$.

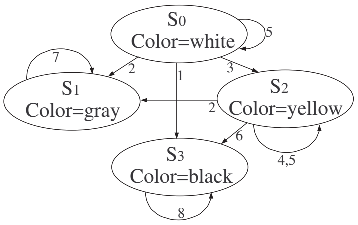

Figure 4. *The state transition diagram for the first phase of Algorithm II for any node.*

1. u is the leader and is in state S_0. u will broadcast message $< \mathit{dominator}(u, u, 0) >$ to its neighbors and go to state S_3.

2. u is in state S_0 or S_2 and receives $< \mathit{dominator}(v, x, l) >$ from neighbor v. Here x is either the leader or null. u will broadcast $< \mathit{dominatee}(u, v, l) >$ and go to S_1.

3. u is in state S_0 and receives the message $< \mathit{dominatee}(v, d, l) >$. u will update d^* to $d^* - 1$, removes v from W_u if $v \in W_u$, set rank_u to $l + 1$, broadcast $< \mathit{active}(u) >$, and then go to S_2.

4. In S_2, u will keep track of all $< \mathit{dominatee}(v, d, l) >$ messages broadcasted in $N_1[u]$ and remove those dominatees in W_u. u updates d^* and rank_u accordingly ($\mathit{rank}_u = 1 + \mathit{max}_v\{l\}$). If there is no broadcasting in $N_1[u]$ for T_0 time unit, u will broadcast message $< \mathit{degree}(u, d^*) >$ if d^* is changed after last broadcasting of $< \mathit{degree} >$ message.

5. u is in state S_0 or S_2. If u receives message $< \mathit{active}(v) >$, then $W_u = W_u \cup \{v\}$; if u receives message $< \mathit{degree}(v, d^*) >$, u updates the local record of v's effective degree.

6. u is in state S_2 and there is no broadcasting in $N_1[u]$ for T_0 time unit. If u has the biggest (d^*, id) compared with all nodes in W_u, u will broadcast $< \mathit{dominator}(u, \mathit{null}, \mathit{rank}_u) >$ and go to state S_3.

7. u is in state S_1. u keeps track of the number of higher rank black neighbors (number of higher rank dominators in $N_1[u]$, denoted by d_b^*). When all of its neighbors are either in S_1 or in S_3, u broadcasts the message $< \mathit{blackdegree}(u, d_b^*) >$.

8. u is in state S_3. u keeps track of the gray neighbor with a lower rank whose (d_b^*, id) is the biggest, in lexicographic order. This is u's candidate dominator in phase 2.

Remarks: (i) Phase 1 generates many stars. Each star consists of one black node, which serves as the center, and many gray nodes dominated by the center. Each gray node in a star sets its dom to the center. All nodes in a star have the same rank, which is also the rank of the star. The dominator of a gray node is its first black neighbor. In other words, a gray node u resides in the star centered at the black neighbor which is the first to go to state S_3. (ii) The star centered at the leader has rank 0. The rank of any other star s centered at u is one plus the highest rank of u's gray neighbors not in s. (iii) We can also understand phase 1 in the following way: phase 1 contains multiple steps. Each step generates a star. At each step except the first one, an active node (in S_2) u with the maximum effective degree compared with all of its active yellow neighbors is colored black. All of u's 1-hop white/yellow neighbors (in state S_1 or S_2) are colored gray. All of u's 2-hop white neighbors (in state S_0) will go to state S_2. (iv) $< degree >$ message is used to announce the number of white/yellow neighbors of node u to all of u's white/yellow neighbors. With this information, a dominator will be elected in next step. $< blackdegree >$ message is used by gray node u to announce the number of black neighbors whose rank is higher than u to all higher rank black neighbors of u. With this information, a black node v can select a gray neighbor with a lower rank than v that connects with many stars.

After phase 1, each node is either in state S_1 or state S_3. A node in state S_1 has a dominator in S_3 while a node in S_3 does not have any dominator (except the leader). Phase 2 will designate a dominator for each black node u (in S_3). u's dominator will be the gray node v in S_1 such that $rank(v) < rank(u)$ and v has the largest (d_b^*, id) among all lower-ranked gray neighbors of u. The transition diagram for phase 2 is shown in Figure 5. We elaborate the details below.

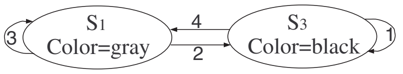

Figure 5. *The state transition diagram for the second phase of Algorithm II for each node.*

1. After a black node u (in S_3) received $< blackdegree >$ from all of its lower-ranked gray neighbors, it broadcasts message $< dominator(u, dom_u, rank_u) >$, where dom_u is the lower-ranked gray neighbor with the highest (d_b^*, id). Here $rank_u$ was set in phase 1.

2. After a gray node u (in S_1) received $< dominator(v, u, r_v) >$ from v, it will broadcast $< dominator(u, dom_u, rank_u) >$ and go to state S_3. Here the dom_u and $rank_u$ parameters for u were set in phase 1.

3. A gray node u in S_1 will keep track of the black neighbor v with the lowest rank by listening to all $< dominator >$ broadcastings. After all black neighbors (generated in phase 1) determine their dominators, u will broadcast message $< dominatee(u, v, rank_v) >$ if v is different from its original dominator dom_u. Note that here u will be a dominatee after phase 2.

4. If u is in state S_3 but no gray node selects u as the dominator, u will broadcast message $< dominatee(u, dom_u, rank_u) >$ and go to state S_1.

Remarks: (i) Phase 2 assigns a dominator v to the center u of each star generated in phase 1 except the star centered at the leader. v is a gray node satisfying the following conditions: v is located in a star with a rank lower than u and v is adjacent to the maximum number of stars with a higher rank than v. (ii) After phase 2, the dominator of a gray node u is always the black neighbor with the lowest rank. (iii) Phase 2 is a local process which occurs among adjacent stars with different ranks. After a black node u received $< blackdegree >$ from all of its lower-ranked gray neighbors, it can select its dominator and broadcast message $< dominator(u, dom_u, rank_u) >$. This means a black node can initiate phase 2 immediately after all necessary information is available. There is no explicit start time for phase 2. Phase 2 is used to connect all stars.

5.2 Performance analysis

In this subsection, we study the performance of Algorithm II.

Lemma 5.1. *In phase 2, the gray node with the maximum (d_b^*, id) will connect with d_b^* number of lower-ranked stars.*

Proof. Let u be the gray node with the maximum (d_b^*, id). Phase 1 ensures that u belongs to the star whose center is the first black neighbor of u. For all other black neighbors (centers of some stars) of u, their ranks must be greater than that of u since a black node in phase 1 always assigns its rank to be one plus the rank of its highest rank dominatee neighbor. Thus all these d_b^* nodes will select u as their dominator in phase 2. In other words, u connects with d_b^* number of lower-ranked stars. ■

Theorem 5.1. *The connected dominating set generated in Algorithm II has a size of at most $(17\frac{1}{3}n_k + 9\frac{2}{3})opt - 2$, where opt is the size of any optimal MCDS for the given instance.*

Proof. Phase 1 computes a MIS. Let A be this MIS with a size $|A|$. From Theorem 3.1, $|A| \leq (4\frac{5}{6}+8\frac{2}{3}n_k)opt$. Note that in phase 2, at most $|A|-1$ nodes in state S_1 will go to state S_3. Now we consider two cases here.

First, if there exists a gray vertice with $d_b^* \geq 2$ at the beginning of phase 2, from Lemma 5.1, the gray vertex u with the maximum (d_b^*, id) will connect d_b^* stars to the higher rank star u resides in phase 2. Therefor the number of nodes changing state from S_1 to S_3 in phase 2 is at most $|A|-2$. Thus the total number of nodes in state S_3 is at most $|A|+|A|-2 \leq (4\frac{5}{6}+8\frac{2}{3}n_k)opt + (4\frac{5}{6}+8\frac{2}{3}n_k)opt - 2 \leq (17\frac{1}{3}n_k + 9\frac{2}{3})opt - 2$.

Secondly, if all gray vertices have $d_b^* \leq 1$ at the beginning of phase 2, then the number of nodes changing state from S_1 to S_3 in phase 2 is exactly $|A|-1$. Since the dominator of any gray vertex u is its first black neighbor, and all other black neighbors of u have a higher rank than u, thus the total number of black neighbors u has is at most 2. In other words, any node in an optimal MCDS is either in A or adjacent to at most 2 vertices in A and any vertex in A is dominated by a vertex in the MCDS. Therefor in this case, $|A| \leq 2 \cdot opt$. Thus the total number of black nodes will be $2\cdot opt + 2\cdot opt - 1 < 4\cdot opt$. ■

Theorem 5.2. *Algorithm II has message complexity $O(n\Delta)$, where n is the total number of vertices and Δ is the maximum node degree.*

Proof. In Algorithm II, the message complexity is dominated by the $<$ $degree$ $>$ messages broadcasted by white vertices in S_2 in phase 1 since each node broadcasts each of the other messages at most twice. Therefore the message complexity is $O(n\Delta)$. ■

6 Conclusion

In this paper, we studied the problem of constructing minimum connected dominating set in geometric k-disk graphs. We first derived the upper bound of the size of any maximal independent set compared to that of a MCDS. Then we proposed two distributed approximation algorithms and studied their performance theoretically.

As a future research activity, we intend to improve the upper bound and design better approximation algorithms for MCDS in geometric k-disk graphs. Note that a polynomial time approximation scheme (PTAS) exists for unit-disk graphs [5] but whether a PTAS exists or not for the general geometric k-disk graphs is still open. This is another problem we intend to target in the future.

References

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *HICSS '02*, volume 9, page 297, 2002.

[2] K. M. Alzoubi, P.-J. Wan, and O. Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *MobiHoc '02*, pages 157–164, 2002.

[3] M. Cardei, M. X. Cheng, X. Cheng, and D.-Z. Du. Connected domination in ad hoc wireless networks. In *CS&I 2002*, pages 251–255, 2002.

[4] X. Cheng, M. Ding, D. H. Du, and X. Jia. Virtual backbone construction in multihop ad hoc wireless networks. *Wireless Communications and Mobile Computing*, 6:183–190, 2006.

[5] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.

[6] I. Cidon and O. Mokryn. Propagation and leader election in a multihop broadcast environment. In *DISC '98*, pages 104–118, 1998.

[7] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

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

[9] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[10] M. D. J. Blum and X. Cheng. *Handbook of Combinatorial Optimization*, chapter Applications of Connected Dominating Sets in Wireless Networks, pages 329–369. Kluwer Academic Publisher, 2004.

[11] C.-Y. Koo, V. Bhandari, J. Katz, and N. H. Vaidya. Reliable broadcast in radio networks: the bounded collision case. In *PODC '06*, pages 258–264, 2006.

[12] Y. Li, M. T. Thai, F. Wang, C.-W. Yi, P.-J. Wan, and D.-Z. Du. On greedy construction of connected dominating sets in wireless networks: Research articles. *Wirel. Commun. Mob. Comput.*, 5(8):927–932, 2005.

[13] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and K.-I. Ko. A greedy approximation for minimum connected dominating sets. *Theor. Comput. Sci.*, 329(1-3):325–330, 2004.

[14] M. T. Thai, F. Wang, D. Liu, S. Zhu, and D.-Z. Du. Connected dominating sets in wireless networks with different transmission ranges. *IEEE Transactions on Mobile Computing*, 6(7):721–730, 2007.

[15] P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *INFOCOM 2002*, pages 1597–1604, 2002.

[16] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *DIALM '99:*, pages 7–14, 1999.

[17] W. Wu, H. Du, X. Jia, Y. Li, and S. C.-H. Huang. Minimum connected dominating sets and maximal independent sets in unit disk graphs. *Theor. Comput. Sci.*, 352(1):1–7, 2006.