A Polynomial-Time Approximation Scheme for the Minimum-Connected Dominating Set in Ad Hoc Wireless Networks

Xiuzhen Cheng
Department of Computer Science, George Washington University, Washington, DC 20052

Xiao Huang
3M Center, Building 0235-03-F-08, St. Paul, MN 55144

Deying Li
Department of Computer Science, City University of Hong Kong, Hong Kong, China

Weili Wu
Department of Computer Science, University of Texas at Dallas, Richardson, Texas 75083

Ding-Zhu Du
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, Minnesota 55455

A connected dominating set in a graph is a subset of vertices such that every vertex is either in the subset or adjacent to a vertex in the subset and the subgraph induced by the subset is connected. A minimum-connected dominating set is such a vertex subset with minimum cardinality. An application in ad hoc wireless networks requires the study of the minimum-connected dominating set in unit-disk graphs. In this paper, we design a (1 + 1/*s*)-approximation for the minimum-connected dominating set in unit-disk graphs, running in time $n^{O((s \log s)^2)}$. © 2003 Wiley Periodicals, Inc.

Keywords: connected dominating set; unit disk graph; polynomial-time approximation scheme; partition

1. INTRODUCTION

Ad hoc wireless networking has attracted more and more attention recently [2, 6, 8, 12]. It will revolutionize information gathering and processing in both urban environments and inhospitable terrain. An ad hoc wireless network is an autonomous system consisting of mobile hosts (or routers) connected by wireless links. It can be quickly and widely deployed. Example applications of ad hoc wireless networks include emergency search-and-rescue operations, decision making in the battlefield, data acquisition operations in inhospitable terrain, etc.

Two important features of an ad hoc wireless network are its *dynamic topology* and *resource limitation.* In an ad hoc wireless network, every host can move in any direction at any time and any speed. There is no fixed infrastructure and central administration. A temporary infrastructure can be formed in any way. Due to multipath fading, multiple access, background noise, and interference from other transmissions, an active link between two hosts may become invalid abruptly. Thus, the communication link is unreliable and retransmission is quite often necessary for reliable services. The resource constraints for an ad hoc wireless network include battery capacity, bandwidth, CPU speed, etc. These two features make routing decisions very challenging.

Existing routing protocols rely on *flooding* for the dissemination of topology update packets (*proactive routing protocols* [5]) or route request packets (*reactive routing protocols* [9, 13]). Networkwide flooding (global flooding) may cause the following two problems:

- *Broadcast storm problem* [12]. Network-wide flooding may result in excessive *redundancy, contention,* and *col-*

Received February 2002; accepted July 2003.
Correspondence to: D.-Z. Du; e-mail: dzd@cs.umn.edu

lision. This causes high protocol overhead and interference with other ongoing communication traffic.

- *Flooding is unreliable* [8]. "In moderately sparse graphs the expected number of nodes in the network that will receive a broadcast message was shown to be as low as 80%" [14].

To overcome or, at least, alleviate these problems, a *virtual backbone-based routing* strategy has been introduced [2, 6, 16]). The most important benefit of virtual backbone-based routing is the dramatic reduction of protocol overhead; thus, it greatly improves the network throughput. This is achieved by propagating control packets inside the virtual backbone, not the whole network. Other benefits include the support of broadcast/multicast traffic and the propagation of "link quality" information for QoS routing [15].

Based on these applications, we can summarize the essential requirements for a virtual backbone as follows: (i) The number of hosts in the backbone is minimized; (ii) all hosts in the backbone are connected; and (iii) each of the hosts not in the backbone has at least one neighbor in the backbone. This is clearly the idea of a minimum-connected dominating set. A *connected dominating set* on a graph is a subset of vertices such that (a) every vertex is either in the subset or adjacent to a vertex in the subset and (b) the subgraph induced by the subset is connected. The problem of a minimum-connected dominating set (MCDS) is to compute a connected dominating set of minimum cardinality.

On the other hand, we assume that an ad hoc wireless network contains only homogeneous mobile hosts. Each host is supplied with an equal-power omnidirectional antenna. Similar assumptions are taken by most researchers in the field of mobile ad hoc wireless networking. Thus, the footprint of an ad hoc wireless network is a unit-disk graph. Indeed, in a *unit-disk graph,* the vertex set consists of a finite number of points in the Euclidean plane and an edge exists between two vertices (points) if and only if the distance between them is at most one.

According to the above analysis, we formulate the problem of constructing a virtual backbone as the problem of *an MCDS in unit-disk graphs.*

The MCDS in general graphs was studied in [7], which proposed a reduction from the set-cover problem. This implies that, for any fixed $0 < \epsilon < 1$, no polynomial time algorithm with performance ratio $(1 - \epsilon)H(\Delta)$ exists unless $NP \subset DTIME[n^{O(\log\log n)}]$ [10], where Δ is the maximum degree and H is the harmonic function. The MCDS in unit-disk graphs is still NP-hard [4]. The best-known performance ratio of previous polynomial-time approximations is a constant > 7 [1, 3, 11]. In this paper, we will propose a Polynomial Time Approximation Scheme (PTAS) for the MCDS in unit-disk graphs.

An algorithm A is a PTAS for a minimization problem with optimal cost OPT if the following is true: Given an instance I of the problem and a small positive error parameter ϵ, (i) the algorithm outputs a solution with cost at most $(1 + \epsilon)OPT$, and (ii) when ϵ is fixed, the running time is bounded by a polynomial in the size of the instance I. If there exists a PTAS for an optimization problem, the problem's instance can be approximated to any required degree.

2. PRELIMINARIES

A *dominating set* in a graph is a subset of vertices such that every vertex is either in the subset or adjacent to at least one vertex in the subset. If, in addition, the subgraph induced by a dominating set is connected, then the dominating set is called a *connected* dominating set. The following is a well-known fact about dominating sets and connected dominating sets:

Lemma 2.1. *For any dominating set D in a connected graph, we can find at most $2(|D| - 1)$ vertices to connect D. Moreover, if D_1^* and D_2^* are, respectively, a minimum dominating set and an MCDS, then $|D_2^*| \leq 3|D_1^*| - 2$.*

We are interested in the minimum-connected dominating set in unit-disk graphs. The unit-disk graph has the following property:

Lemma 2.2. *Suppose that a unit-disk graph G lies in an $m \times m$ square such that every vertex is away from the boundary with distance at least 1/2. Then, G has at most $\lfloor 4m^2/\pi \rfloor$ connected components.*

Proof. Let x denote the number of connected components of such a unit-disk graph. From each connected component, we choose a vertex and identify it with the center of a unit-disk. (A unit-disk has diameter one.) Such unit-disks are disjoint and all lie in the cell. Therefore, we have

$$x \cdot \pi(1/2)^2 < m^2.$$

Hence, $x < 4m^2/\pi$. ∎

It has been known that the MCDS has some polynomial-time approximation with a constant performance ratio [1, 3, 11]. Here, we quote a result from [3]:

Lemma 2.3. *There exists a polynomial-time approximation for the MCDS in unit-disk graphs, with performance ratio eight.*

In design of a PTAS for the MCDS, we sometimes consider the following generalization of the concept of dominating set and connected dominating set:

Consider a graph $G = (V, E)$. Suppose that H is a subgraph of G. A subset D of vertices in G is said to be a *connected dominating set* in G for H if every vertex in H is either in D or adjacent to a vertex in D and, in addition, the subgraph of G induced by D is connected.

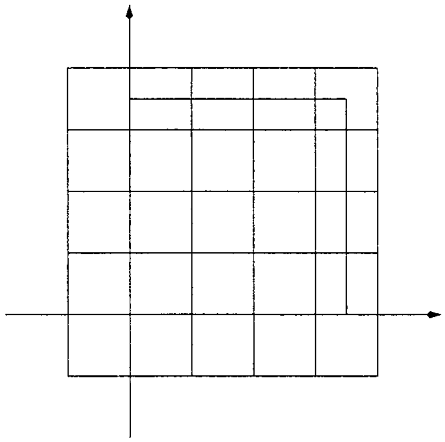

FIG. 1. Squares Q and $\bar{Q}$.

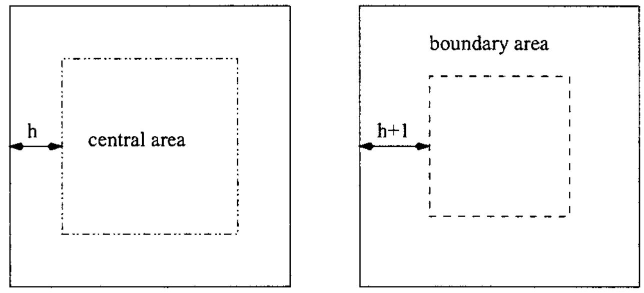

FIG. 2. Central area and boundary area.

3. MAIN RESULTS

In this section, we will construct a PTAS for the MCDS in unit-disk graphs. The general picture of this construction is as follows: First, we divide a space, containing all vertices of the input unit-disk graphs, into a grid of small cells. For each small cell, take the points h distance away from the boundary (the central area of the cell). Then, we optimally compute a minimum union of connected dominating sets in each cell for connected components of the central area of the cell. The key lemma is to prove that the union of all such minimum unions is no more than the MCDS for the whole graph. Then, for vertices not in central areas, just use the part of an 8-approximation lying in boundary areas (within distance $h + 1$ away from the boundary, with some overlap with the central areas) to dominate them. This part, together with the above union, forms a connected dominating set for the whole input unit-disk graph. Finally, using the shifting argument (i.e., shift the grid around to get partitions at different coordinates) to make sure there are at least half good partitions having good approximations overall.

Next, we work out details along the above lines:

For the input connected unit-disk graph $G = (V, E)$, we initially find a minimal square Q containing all vertices in V. Without loss of generality, assume that $Q = \{(x, y) | 0 \leq x \leq q, 0 \leq y \leq q\}$. Let m be a large integer that we will determine later. Let $p = \lfloor q/m \rfloor + 1$. Consider the square $\bar{Q} = \{(x, y) | -m \leq x \leq mp, -m \leq y \leq mp\}$. Partition $\bar{Q}$ into a $(p + 1) \times (p + 1)$ grid so that each cell is an $m \times m$ square excluding the top and the right boundaries and, hence, no two cells are overlapping each other. This partition of $\bar{Q}$ is denoted by $P(0, 0)$ (Fig. 1). In general, the partition $P(a, b)$ is obtained from $P(0, 0)$ by shifting the bottom-left corner of $\bar{Q}$ from $(-m, -m)$ to $(-m + a, -m + b)$.

For each cell e as an $m \times m$ square, we denote by $C_e(d)$ the set of points in e away from the boundary by distance at least d, for example, $C_e(0)$ is the cell e itself. Fix a positive integer h whose value will be determined later. We will call $C_e(h)$ the *central area* of e and $C_e(0) - C_e(h + 1)$ the *boundary area* of e (Fig. 2). For simplicity of notation, we denote $B_e(d) = C_e(0) - C_e(d)$. Note that, for each cell, its boundary area and central area are overlapping with the width one. For each partition $P(a, a)$, denote by $C^a(d)$ ($B^a(d)$) the union of $C_e(d)(B_e(d))$ for e over all cells in $P(a, a)$. $C^a(h)$ and $B^a(h + 1)$ are called the *central area* and the *boundary area* of $P(a, a)$.

For a graph G, denote by $G_e(d)(\tilde{G}_e(d))$ the subgraph of G induced by all vertices lying in $C_e(d)(B_e(d))$ and by $G^a(d)(\tilde{G}^a(d))$ the subgraph of G induced by all vertices lying in $C^a(d)(B^a(d))$.

Let $G = (V, E)$ be an input connected unit-disk graph. Consider a subgraph $G_e(h)$. This subgraph may consist of several connected components. Let K_e be a dominating set in $G_e(0)$ for $G_e(h)$ with minimum cardinality such that, for each connected component H of $G_e(h)$, K_e contains a connected component dominating H. In other words, K_e is a minimum union of connected dominating sets in $G_e(0)$ for connected components of $G_e(h)$. Now, we denote by K^a the union of K_e for e over all cells in partition $P(a, a)$.

By Lemma 2.3, we can compute, in polynomial time, a connected dominating set F for an input connected graph G within a factor of 8 from optimal. Set $A^a = K^a \cup \tilde{F}^a(h + 1)$. [Note that we consider F as a graph without edges. According to the above definition, $\tilde{F}^a(h + 1) = F \cap B^a(h + 1)$.]

Lemma 3.1. *For $0 \leq a \leq m - 1$, A^a is a connected dominating set for input graph G. Moreover, A^a can be computed in time $n^{O(m^2)}$.*

Proof. A^a is clearly a dominating set for input graph G. We next show its connectivity. Note that for any connected component H of the subgraph $G_e(h)$ for some cell e in partition $P(a, a)$, if a connected component E of $\tilde{F}^a(h + 1)$ has a vertex in H, then E must connect to the connected dominating set D_H for H. This means that D_H has been making up the connections of F lost from cutting a part in H. Therefore, the connectivity of A^a follows from the connectivity of F.

To establish the time for computing A^a, we note the fact that, for a square with edge length $\sqrt{2}/2$, all vertices lying inside the square induce a complete subgraph in which any vertex must dominate all other vertices. It follows from this

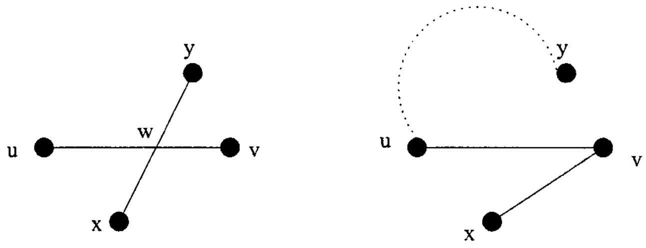

FIG. 3. Two edges (u, v) and (x, y) have crosspoint w.

fact that the minimum dominating set for the subset V_e of vertices lying in cell e has size $\le (\lceil \sqrt{2}m \rceil)^2$. Hence, the MCDS for V_e has size at most $3(\lceil \sqrt{2}m \rceil)^2$ by Lemma 2.1. Therefore, $|K_e| \le 3(\lceil \sqrt{2}m \rceil)^2$. Suppose that cell e contains n_e vertices of the input unit-disk graph. Then, the number of candidates for each dominator in K_e is at most

$$\sum_{k=0}^{3(\lceil \sqrt{2}m \rceil)^2} \binom{n_e}{k} = n_e^{O(m^2)}.$$

Hence, computing A^a can be done in time

$$\sum_e n_e^{O(m^2)} \le \left(\sum_e n_e \right)^{O(m^2)} = n^{O(m^2)}.$$

■

By Lemma 3.1, we may take A^a to approximate the MCDS. The next lemma will help us estimate the approximation performance of A^a:

Lemma 3.2. *Suppose that* $h = 7 + 3\lfloor \log_2(4m^2/\pi) \rfloor$. *Let* D^* *be an MCDS for input graph G. Then,* $|K^a| \le |D^*|$ *for* $0 \le a \le m - 1$.

Proof. Recall that $G^a(h)$ is the subgraph of input graph $G = (V, E)$ induced by its vertices lying in the central area $C^a(h)$ of the partition $P(a, a)$. Let D be an MCDS in G for $G^a(h)$. Then, we must have $|D| \le |D^*|$.

Now, let $G[D]$ denote the subgraph of G induced by D. We first claim that $G[D]$ has a spanning tree T without crossing edges in the plane. In fact, suppose that T is a spanning tree of $G[D]$ with the minimum total edge length. Suppose that T contains two edges (u, v) and (x, y) crossing at a point w in the plane. Without loss of generality, assume that segment (v, w) is the shortest one among the four segments (u, w), (v, w), (x, w), and (y, w) (Fig. 3). Removal of (x, y) from T would break T into two connected components containing vertices x and y, respectively. One of them contains edge (u, v). Note that $d(x, v) < d(x, w) + d(v, w) \le d(x, w) + d(w, y) \le 1$ and $d(y, v) < d(y, w) + d(v, w) \le d(y, w) + d(w, x) \le 1$. Therefore, we can add either (x, v) or (y, v) to connect the two connected components of $T - (x, y)$ into one. [In Fig. 3, the right side shows a case where (u, v) is in the connected component containing y and, hence, (v, x) is added to connect the two components into one.] This operation reduces the total edge length of the tree, contradicting the assumption on T.

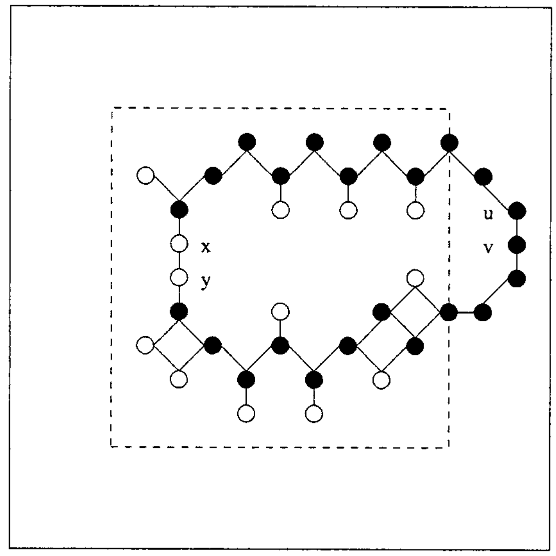

FIG. 4. Operation 2.

Assume that T is a spanning tree of $G[D]$ without any crosspoint. Let T_b be the subforest of T induced by those vertices not dominating any vertex in $G^a(h)$. We next modify T to a forest with three operations:

Operation 1: If, after deleting a vertex u of T_b, T still keeps the following property (B1), then delete u.

(B1) For any connected component H of $G^a(h)$, T connects every two vertices in $H \cap T$, that is, T has a connected component dominating H.

Operation 2: If, after deleting an edge of T, T still keeps the property (B1), then delete the edge (Fig. 4).

Through Operations 1 and 2, T becomes a forest with the property that deleting any vertex or edge would destroy property (B1). Now, we apply the third operation to T.

Operation 3: If T_b has two adjacent vertices u and v both with degree two, then delete them and restore the property (B1) as follows: Note that deleting u and v breaks a connected component of T into two parts, say C_1 and C_2. Since T already passed Operation 1, there must exist a connected component H of $G_e(h)$ such that $T \cap H$ exists in both C_1 and C_2. Since $T \cap C_1$ and $T \cap C_2$ dominate H, there must exist either one vertex x in H such that x is dominated by both $T \cap C_1$ and $T \cap C_2$ or two adjacent vertices x and y in H such that x is dominated by $T \cap C_1$ and y is dominated by $T \cap C_2$. Therefore, adding either x or x and y to T would restore the property (B1).

After Operation 3 is employed once, it may be possible to apply Operations 1 and 2 again. At any time, if Operation 1 or 2 can be applied, then we use it; if Operations 1 and 2 cannot be applied but Operation 3 can be, then we employ Operation 3. Since both Operations 1 and 3 reduce the number of vertices in T_b and Operation 2 reduces the

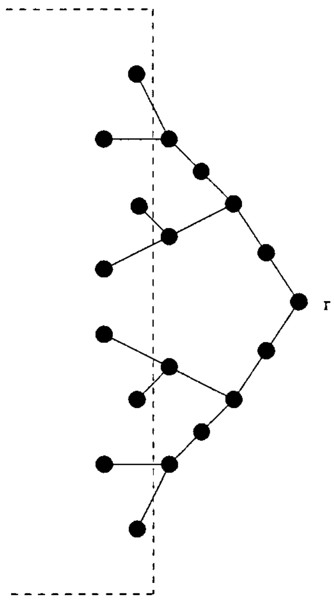

FIG. 5. Tree T'.

number of edges of T without increasing the number of vertices of T_b, this process has to end in finitely many steps. At the end, forest T would still have property (B1) and in addition have the following properties:

(B2) T_b has no adjacent two vertices both with degree two.

(B3) T has at most $|D^*|$ vertices. (Note that initially T is a spanning tree for $G[D]$ and $|D| \leq |D^*|$. Since Operations 1, 2, and 3 do not increase the number of vertices in T, the final T has at most $|D^*|$ vertices.)

Next, we move some edges of T to the inside of $C^a(h - 1)$ by the following operation:

Operation 4: If T has two vertices u and v lying in $C^a(h - 1)$ such that $d(u, v) \leq 1$ and T has a path between u and v which contains an edge not lying in $C^a(h - 1)$, then delete the edge and add edge (u, v).

After Operation 4 is employed once, it may be possible to apply Operations 1, 2, and 3 again. At any time, if Operation 1, 2, or 3 can be applied, then we use it; if Operations 1, 2, and 3 cannot be applied but Operation 4 can be, then we employ Operation 4. Since Operation 4 reduces the number of edges of T not lying in $C^a(h - 1)$ and Operations 1, 2, and 3 do not increase the number of edges of T not lying in $C^a(h - 1)$, this process has to end in finitely many steps. At the end, forest T would still have properties (B1), (B2), and (B3) and, in addition, have the following property:

(B4) Operation 4 cannot be applied.

Since any vertex dominating some vertex in the central area of a cell e must lie in $C_e(h - 1)$, every vertex of T lying in $B_e(h - 1)$ must belong to T_b.

Now, we consider a maximal subtree T' of T such that

(C) T' has all leaves in $C_e(h - 1)$ and all other vertices not in $C_e(h - 1)$ (Fig. 5).

For simplicity, we call such a maximal subtree satisfying (C) an *ear* of the central area of cell e. We claim that T' lies in cell e. To show our claim, suppose that T' has k leaves. Since $h \geq 7$, every edge of T' incident to a leaf does not lie in $C^a(h - 1)$. Thus, every path in T' connecting two leaves must contain an edge not lying in $C^a(h - 1)$. By (B4), every two leaves of T' have distance more than one. By Lemma 2.2, $k \geq \lfloor 4m^2/\pi \rfloor$. Please recall the notation that $T_e(h - 1)$ is the subgraph of T induced by its vertices lying in the area $C_e(h - 1)$ and T is a forest obtained in the above proof. Note that $T'_e(h - 1)$ consists of k leaves of T' and, hence, has k connected components. The *outer path* p of T' is a path between two leaves such that T' lies in the area between the path p and the boundary of $C_e(h)$. Since T has no crosspoint and T' is a maximal subtree satisfying (C), only vertices in path p may meet an edge in T but not in T'.

For contradiction, suppose that T' has a vertex r lying outside of cell e. Without loss of generality, we may assume that r is on the path p. We consider r as a root for T' and study the k paths from leaves to r. The path p is broken at r into two such paths. Note that any path passing through area $C_e(h - 4) - C_e(h - 1)$ must meet an edge not on the path. (Otherwise, the path would contain two vertices in T_b both with degree two.) It follows that, except for the two paths obtained from path p, every path has to be merged into another one in area $C_e(h - 4) - C_e(h - 1)$. This means that these k paths become at most $2 + k/2$ paths when they go out from $C_e(h - 4)$, namely, $T'_e(h - 4)$ contains at most $2 + (k - 2)/2$ connected components. Similarly, $T'_e(h - 1 - 3(\lfloor \log_2(k - 2) \rfloor))(\subseteq T'_e(6))$ contains at most three connected components and $T'_e(3)$ contains at most two connected components, that is, all k paths in $C_e(3)$ have merged into two paths. Note that these two paths will merge into one at r lying outside cell e. Therefore, each of them has a vertex u in area $C_e(0) - C_e(3)$ incident to an edge in $T - T'$. This means that there must exist another cell e' whose central area has an ear T'' touching T' at a point in cell e. (Note that after Operations 1, 2, and 3 T_b is contained in the union of ears of all central areas of cells.) So, T'' cannot lie in cell e'.

Similarly, this implies the existence of another cell e'' whose central area has an ear T''' touching T'' at a point in cell e'. Since T contains no cycle, this process may go on indefinitely, so that a path of infinite length is found in T, a contradiction. This contraction completes the proof of our claim that T' lies in cell e.

By (B1) and our claim, $T_e(0)$ is a union of connected dominating sets for connected components of $G_e(h)$. It follows that the number of vertices in $T_e(0)$ is at least $|K_e|$ since K_e is a minimum one. Thus,

$$|K^a| = \sum_e |K_e| \leq \sum_e |T_e(0)| \leq |T| \leq |D^*|,$$

where $|T|$ denotes the number of vertices in T. ∎

We are ready to present the following main theorem:

Theorem 3.3. *Suppose that $h = 7 + 3\lfloor \log_2(4m^2/\pi) \rfloor$ and $\lfloor m/(h+1) \rfloor \geq 32s$. Then, there is at least half of $i = 0, 1, \ldots, \lfloor m/(h+1) \rfloor - 1$ such that $A^{i(h+1)}$ is a $(1 + 1/s)$-approximation for the minimum connected dominating set.*

Proof. By Lemma 3.2, for every $i = 0, 1, \ldots, \lfloor m/(h+1) \rfloor - 1$, $|K^{i(h+1)}| \leq |D^*|$, where D^* is an MCDS for G. Recall that F is a connected dominating set for G such that $|F| \leq 8|D^*|$ and $\tilde{F}^a(h+1) = F \cap B^a(h+1)$. Moreover, let $F_H^a(F_V^a)$ denote the subset of vertices in $\tilde{F}^a(h+1)$ each with distance $< h + 1$ from the horizontal (vertical) boundary of some cell in $P(a, a)$. Then, $\tilde{F}^a(h+1) = F_H^a \cup F_V^a$. Moreover, all $F_H^{i(h+1)}$ for $i = 0, 1, \ldots, \lfloor m/(h+1) \rfloor - 1$ are disjoint. Hence,

$$\sum_{i=0}^{\lfloor m/(h+1) \rfloor - 1} |F_H^{i(h+1)}| \leq |F| \leq 8|D^*|.$$

Similarly, all $F_V^{i(h+1)}$ for $i = 0, 1, \ldots, \lfloor m/(h+1) \rfloor - 1$ are disjoint and

$$\sum_{i=0}^{\lfloor m/(h+1) \rfloor - 1} |F_V^{i(h+1)}| \leq |F| \leq 8|D^*|.$$

Thus,

$$\sum_{i=0}^{\lfloor m/(h+1) \rfloor - 1} |\tilde{F}^{i(h+1)}(h+1)| \leq \sum_{i=0}^{\lfloor m/(h+1) \rfloor - 1} (|F_H^{i(h+1)}| + |F_V^{i(h+1)}|) \leq 16|D^*|.$$

Therefore,

$$\sum_{i=0}^{\lfloor m/(h+1) \rfloor - 1} |A^{i(h+1)}| \leq \sum_{i=0}^{\lfloor m/(h+1) \rfloor - 1} (|K^{i(h+1)}| + |\tilde{F}^{i(h+1)}(h+1)|) \leq (\lfloor m/(h+1) \rfloor + 16)|D^*|,$$

that is,

$$\frac{1}{\lfloor m/(h+1) \rfloor} \sum_{i=0}^{\lfloor m/(h+1) \rfloor - 1} |A^{i(h+1)}| \leq (1 + 1/(2s))|D^*|.$$

This means that there are at least half of $A^{i(h+1)}$ for $i = 0, 1, \lfloor m/(h+1) \rfloor - 1$ satisfying

$$|A^{i(h+1)}| \leq (1 + 1/s)|D^*|.$$

The following corollary follows immediately from the theorem:

Corollary 3.4. *There is a $(1 + 1/s)$-approximation for an MCDS in connected unit-disk graphs, running in time $n^{O((s \log s)^2)}$.*

Proof. Note that computing each A^a needs time $n^{O(m^2)}$. By Theorem 3.3, a $(1 + 1)/s)$-approximation can be obtained by computing all $\lfloor m/(h+1) \rfloor$ A^a's and choosing the best one. Thus, the total running time is $mn^{O(m^2)} = n^{O(m^2)}$. Choose m to be the least integer satisfying $\lfloor m/(h+1) \rfloor \geq 32s$, where $h = 7 + 3\lfloor \log_2(4m^2/\pi) \rfloor$. Then, $m = O(s \log s)$. This completes the proof. ■

4. CONCLUSIONS

We have designed a PTAS for the MCDS in unit-disk graphs. There is evidence to show that currently existing implemented approximations have a large space for improvement.

Acknowledgments

The authors wish to thank Dr. Penjun Wan for his helpful suggestions and corrections and also wish to thank a referee for insightful comments.

REFERENCES

[1] K.M. Alzoubi, P.-J. Wan, and O. Frieder, New distributed algorithm for connected dominating set in wireless ad hoc networks, Proc HICSS, Hawaii, 2002, pp. 3881–3887.

[2] A.D. Amis and R. Prakash, Load-balancing clusters in wireless ad hoc networks, Proc 3rd IEEE Symp on Application-Specific Systems and Software Engineering Technology, 2000, pp. 25–32.

[3] X. Cheng and D.-Z. Du, Virtual backbone-based routing in ad hoc wireless networks, Technical report 02-002, Department of Computer Science and Engineering, University of Minnesota.

[4] B.N. Clark, C.J. Colbourn, and D.S. Johnson, Unit disk graphs, Discr Math 86 (1990), 165–177.

[5] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, and L. Viennot, Optimized link state routing protocol, IETF Internet Draft, draft-ietf-manet-olsr-05.txt, October 2001.

[6] B. Das and V. Bharghavan, Routing in ad hoc networks using minimum connected dominating sets, ICC '97, Montreal, Canada, June 1997, pp. 376–380.

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

[8] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, Scenario-based performance analysis of routing protocols for mobile ad hoc networks, Proc IEEE MOBICOM, Seattle, August 1999, pp. 195–206.

[9] D.B. Johnson and D.A. Maltz, "Dynamic source routing in ad hoc wireless networks," Mobile computing, Tomasz Imi-

elinski and Hank Korth (Editors), Kluwer, Boston, 1996, pp. 153–181.

[10] C. Lund and M. Yannakakis, On the hardness of approximating minimization problems, J ACM 41 (1994), 960–981.

[11] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi, and D.J. Rosenkrantz, Simple heuristics for unit-disk graphs, Networks 25 (1995), 59–68.

[12] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, The broadcast storm problem in a mobile ad hoc network, Proc MOBICOM, Seattle, 1999, pp. 151–162.

[13] C.E. Perkins and E.M. Royer, Ad hoc on-demand distance vector routing, Proc 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90–100.

[14] P. Sinha, R. Sivakumar, and V. Bharghavan, Enhancing ad hoc routing with dynamic virtual infrastructures, INFOCOM 3 (2001), 1763–1772.

[15] R. Sivakumar, P. Sinha, and V. Bharghavan, CEDAR: A core-extraction distributed ad hoc routing algorithm, IEEE J Sel Areas Commun 17 (1999), 1454–1465.

[16] J. Wu and H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, Proc 3rd Int Workshop on Discrete Algorithms and Methods for MOBILE Computing and Communications, Seattle, WA, 1999, pp. 7–14.