PTAS for Minimum Connected Dominating Set in Unit Ball Graph

Zhao Zhang[1,⋆] Xiaofeng Gao[2], Weili Wu[2,⋆⋆], and Ding-Zhu Du[2]

[1] College of Mathematics and System Sciences, Xinjiang University
Urumqi, Xinjiang, 830046, People's Republic of China

[2] Department of Computer Science, University of Texas at Dallas
Richardson, Texas, 75080, USA

Abstract. When sensors are deployed into a space instead of a plane, the mathematical model for the sensor network should be a unit ball graph instead of a unit disk graph. It has been known that the minimum connected dominating set in unit disk graph has a polynomial time approximation scheme (PTAS). Could we extend the construction of this PTAS for unit disk graphs to unit ball graphs? The answer is NO. In this paper, we will introduce a new construction, which gives not only a PTAS for the minimum connected dominating set in unit ball graph, but also improves running time of PTAS for unit disk graph.

Keywords: wireless sensor network, connected dominating set, unit ball graph.

1 Introduction

Virtual backbone in wireless sensor network has a wide range of applications (cf [3] and references there). A virtual backbone is a subset of nodes D such that non-adjacent nodes can communicate with each other though the nodes in D. Modeling the wireless sensor network as a graph, the virtual backbone is exactly a connected dominating set. A *dominating set* of a graph G is a subset D of vertices such that every vertex x in $V(G) \setminus D$ is adjacent to a vertex y in D. Vertex x is said to be *dominated* by y, or y is said to *dominate* x. A vertex $y \in D$ dominates itself. A *connected dominating set* is a dominating set D such that the subgraph of G induced by D, denoted by $G[D]$, is connected. Because of source limitation, it is often required that the size of the virtual backbone is as small as possible. Hence we are faced with a *minimum connected dominating set problem* (MCDS): to find a connected dominating set with the minimum cardinality. The MCDS has been studied extensively in the literatures [2,11,12,14,15,16,18].

⋆ This research is supported by NSFC (60603003), the Key Project of Chinese Ministry of Education (208161) and Scientific Research Program of the Higher Education Institution of XinJiang. The work was completed when the first author was visiting Department of Computing Science, the University of Texas at Dallas.

⋆⋆ This work is supported in part by the National Science Foundation under grant CCF-0514796 and CNS-0524429.

Y. Li et al. (Eds.): WASA 2008, LNCS 5258, pp. 154–161, 2008.

In practice, the sensors are often assumed to be homogeneous, that is, they have omnidirectional antennas with the same transmission range. In this case, the topology of the 3-dimensional wireless sensor network can be modeled as a unit ball graph. In an *unit ball graph* (UBG), each vertex corresponds to a point in the space, two vertices are adjacent if and only if the Euclidean distance between their corresponding points is less than or equal to one. In another word, a vertex u is adjacent with a vertex v if and only if u is within the transmission range of v, which has been scaled to one. When restricted to the plane, a unit ball graph degenerates to a *unit disk graph* (UDG). Compared with the large number of studies on UDGs, the study on UBGs are relatively much less. However, there are cases in which 3-dimensional models are needed, such as under-water sensor systems, outer-space sensor systems, notebooks in a multi-layered buildings, etc.

For MCDS in general graphs, it was proved in [8] that for any $0 < \rho < 1$, there is no polynomial time $\rho \ln n$-approximation unless $NP \subseteq DTIME(n^{O(\ln n)})$, where n is the number of vertices. A greedy $(\ln \Delta + 3)$-approximation [13] and a greedy $(\ln \Delta + 2)$-approximation [8,13] were given, where Δ is the maximum degree of the graph. When restricted to UDG, the MCDS problem is still NP-hard [7]. Hence computing an MCDS in a UBG is also NP-hard. Distributed constant-approximations for MCDS in UDG were studied in [1,5,10,17], etc. Also by distributed strategy, Butenko and Ursulenko [4] gave a 22-approximation for MCDS in UBG. As to centralized algorithm for CDS in UDG, Cheng et al [6] gave a polynomial time approximation scheme (PTAS), that is, for any $\varepsilon > 0$, there exists a polynomial-time $(1+\varepsilon)$-approximation. The question is: can their method be generalized to obtain a PTAS for MCDS in UBG? The answer is 'no', since their proof depends on a geometrical property which holds in the plane but is no longer true in the space.

In this paper, we present a PTAS for UBG. The method of analyzing the performance ratio is new. In fact, this method can be used to compute CDS for any n-dimensional unit ball graph. Furthermore, when our method is applied to UDG, the running time can be improved, compared with the algorithm presented in [6].

In section 2, the algorithm is presented, the correctness is proved, the time complexity is analyzed. In section 3, we prove that this algorithm is a PTAS. A conclusion is given in section 4.

2 The Algorithm

In this section, we present an algorithm for MCDS in UBG. The algorithm uses partition technique combined with a shifting strategy (which was introduced by Hochbaum and Maass [9]).

Let $Q = \{(x, y, z) \mid 0 \leq x \leq q, 0 \leq y \leq q, 0 \leq z \leq q\}$ be a minimal 3-dimensional cube containing all the unit balls. For a given positive real number $\varepsilon < 1$, let m be an integer with $m = \lceil 300\rho/\varepsilon \rceil$, where ρ is the performance ratio of a constant-approximation for MCDS in UBG, for example $\rho = 22$ by the algorithm given by Butenko and Ursulenko [4]. Set $p = \lfloor q/m \rfloor + 1$, and

$\tilde{Q} = \{(x, y) \mid -m \le x \le mp, -m \le y \le mp, -m \le z \le mp\}$. Divide $\tilde{Q}$ into $(p+1) \times (p+1) \times (p+1)$ grid such that each cell is an $m \times m \times m$ cube. Denote this partition as $P(0)$. For $a = 0, 1, ..., m-1$, $P(a)$ is the partition obtained by shifting $P(0)$ such that the left-bottom-hind corner of $P(a)$ is at the coordinate $(a-m, a-m, a-m)$. For each cell e, the *boundary region* B_e of e is the region contained in e such that each point in this region is at most distance 3 from the boundary of e. The *central region* C_e of e is the region of e such that each point is at least distance 2 away from the boundary of e. Note that B_e and C_e have an overlap.

Algorithm

`Input:` The geometric representation of a connected unit ball graph G and a positive real number $\varepsilon < 1$.

`Output:` A connected dominating set D of G.

1. Let $m = \lceil 300\rho/\varepsilon \rceil$.
2. Use the ρ-approximation algorithm to compute a connected dominating set D_0 of G. For each $a \in \{0, 1, ..., m-1\}$, denote by $D_0(a)$ the set of vertices of D_0 lying in the boundary region of $P(a)$. Choose a^* with the minimum $|D_0(a)|$.
3. For each cell e of $P(a^*)$, denote by G_e the subgraph of G induced by the vertices in the central region C_e. Compute a minimum subset D_e of vertices in e, such that

$$\text{for each component } H \text{ of } G_e,\ G[D_e] \text{ has a connected component dominating } H. \tag{1}$$

4. Let $D = D_0(a^*) \cup \bigcup_{e \in P(a^*)} D_e$.

The following lemma shows the correctness of the algorithm.

Lemma 1. *The output D of the algorithm is a CDS of G.*

Proof. We first show that D is a dominating set. For each vertex $x \in V(G)$, suppose x is in cell e. If $x \in C_e$, then x is dominated by D_e. If $x \in e \setminus C_e$, then x is in the region of e at distance less than two from the boundary of e. If $x \in D_0$, then $x \in D_0(a^*)$. If $x \notin D_0$, then the vertex $y \in D_0$ which dominates x is in $D_0(a^*)$. By the arbitrariness of x, D is a dominating set of G.

Next, we show that $G[D]$ is connected.

Suppose F_1, F_2 are two components of $G[D_0(a^*)]$ which can be connected by D_0 through the central region of some cell e. Then there exist two vertices $x_1 \in V(F_1) \cap B_e \cap C_e$ and $x_2 \in V(F_2) \cap B_e \cap C_e$ such that x_1, x_2 are in a same component H of G_e. By step 3 of the algorithm, x_1 and x_2 are connected through D_e, and thus F_1 and F_2 are also connected through $D_e \subseteq D$. We have shown that any components of $G[D_0(a^*)]$ are connected in $G[D]$.

Let $\tilde{G}$ be the component of $G[D]$ containing all vertices in $D_0(a^*)$. If $\tilde{G} \ne G[D]$, then there exists a cell e and a component R of $G[D_e]$ such that $V(R) \cap D_0(a^*) = \emptyset$ and R is not adjacent with any vertex in $D_0(a^*)$. Let x be a vertex in

D_0 such that x dominates some vertex $y \in V(R)$ (y may coincide with x). Since $x \notin D_0(a^*)$, we have $x \in e \backslash B_e$. Hence $y \in C_e$. Let H be the connected component of G_e containing y. By step 3 of the algorithm, we see that R dominates H. Since $G[D_0]$ is connected, there is a path in $G[D_0]$ connecting x to the other parts of G outside of cell e. Such a path must contain a vertex $z \in D_0 \cap B_e \cap C_e \subseteq D_0(a^*)$. Note that z is also in H. Hence there is a vertex w in $V(R)$ dominating z, contradicting that R is not adjacent with any vertex in $D_0(a^*)$. Hence $\tilde{G} = G[D]$, and thus $G[D]$ is connected. □

The following lemma is a well-known fact about dominating set and connected dominating set.

Lemma 2. *For any dominating set D in a connected graph, at most $2(|D|-1)$ vertices are needed to connect D. In particular, $|D_2| \leq 3|D_1| - 2$, where D_1, D_2 are, respectively, a minimum dominating set and a minimum CDS.*

The next lemma shows that the time complexity of the algorithm is polynomial in n and ε.

Lemma 3. *The above algorithm runs in time $n^{O(1/\varepsilon^3)}$.*

Proof. Clearly, the most time-consuming part is the third step. Since any vertex in a $\sqrt{3}/3 \times \sqrt{3}/3 \times \sqrt{3}/3$ cube dominates any other vertices in the same cube, we see that a minimum dominating set of e uses at most $(\sqrt{3}m)^3$ vertices. By Lemma 2, $|D_e| \leq 3(\sqrt{3}m)^3$. Hence the exhaust search takes time at most $\sum_{k=0}^{(3\sqrt{3}m)^3} \binom{n_e}{k} = n_e^{O(m^3)}$ to compute D_e, where n_e is the number of vertices in e. It follows that the total time complexity is bounded by $\sum_{e \in P(a^*)} n_e^{O(m^3)} = n^{O(m^3)} = n^{O(1/\varepsilon^3)}$. □

3 The Performance Ratio

In this section, we show that our algorithm is a PTAS for CDS in UBG. For this purpose, we need the following two lemmas.

For a path P in G, the length of P, denoted by $len(P)$, is the number of edges in P. Let H be a subgraph of G. For two subgraphs H_1 and H_2 of G, the distance between H_1 and H_2 in H is $dist_H(H_1, H_2) = \{len(P) \mid P$ is a shortest path connecting H_1 and H_2 in $H\}$. In another word, if $dist_H(H_1, H_2) = k$, then H_1 and H_2 can be connected through at most $k-1$ vertices of H. The following lemma can be easily seen from the definition of dominating set.

Lemma 4. *Let H be a connected subgraph of G, and D be a subset of $V(G)$ dominating H. If $G[D]$ does not contain a connected component dominating H, then there exist two components R and K of $G[D]$ such that $dist_H(R, K) \leq 3$.*

The following lemma plays an important role in analyzing the performance ratio of the algorithm.

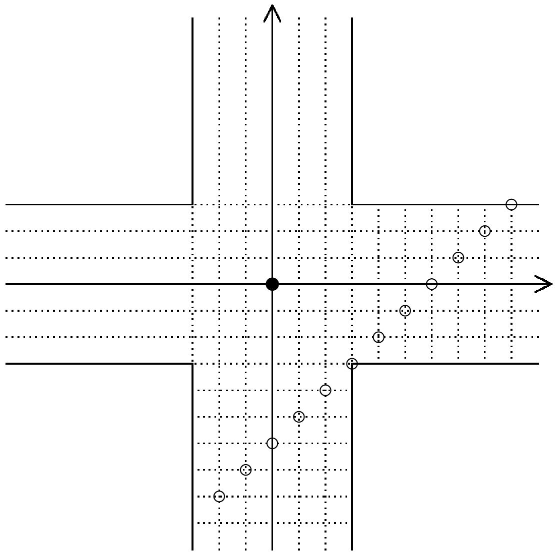

Fig. 1. When the partition shifts, each vertex falls into at most 12 boundary regions

Lemma 5. *For any vertex u in a unit ball graph G, the neighborhood $N_G(u)$ contains at most 12 independent vertices.*

Proof. The result can be obtained by transforming the problem into the famous Gregory-Newton Problem concerning about kissing number [19]. □

Next, we analyze the performance ratio of the algorithm.

Theorem 1. *The algorithm is a $(1+\varepsilon)$-approximation for CDS in UBG.*

Proof. Let D^* be an optimal CDS of G.

Note that when a runs over $0, 1, ..., m-1$, each vertex belongs to at most 12 boundary regions of $P(a)$'s (see Fig. 1). Hence

$$|D_0(0)| + |D_0(1)| + ... + |D_0(m-1)| \le 12|D_0|,$$

and thus

$$|D_0(a^*)| \le \frac{12}{m}|D_0| \le \frac{12\rho}{m}|D^*| \le \frac{\varepsilon}{25}|D^*|. \tag{2}$$

In the following, we are to add some vertices to D^* such that the resulting vertex set $\tilde{D}$ satisfies:

(i) $|\tilde{D}| \le |D^*| + 24|D_0(a^*)|$, and

(ii) for each cell e and each connected component H of G_e, $G[\tilde{D} \cap e]$ contains a connected component dominating H.

Before showing how to construct $\tilde{D}$, we first show that as long as this can be done, then the theorem is proved. In fact, since D_e is a minimum subset of e satisfying the requirement (1) and $\tilde{D} \cap e$ satisfies (ii), we have

$$|D_e| \leq |\tilde{D} \cap e|.$$

Then it follows from condition (i) and inequality (2) that

$$\begin{aligned} |\bigcup_{e \in P(a^*)} D_e| = \textstyle\sum_{e \in P(a^*)} |D_e| \leq \textstyle\sum_{e \in P(a^*)} |\tilde{D} \cap e| \\ = |\tilde{D}| \leq |D^*| + 24|D_0(a^*)| \leq (1 + \tfrac{24\varepsilon}{25})|D^*|. \end{aligned} \tag{3}$$

Combining inequalities (2) and (3), we have

$$|D| \leq |\bigcup_{e \in P(a^*)} D(e)| + |D_0(a^*)| \leq (1 + \varepsilon)|D^*|,$$

where D is the output of the algorithm. This proves the theorem.

In the following we show how to construct $\tilde{D}$ satisfying conditions (i) and (ii).

We first claim that for any cell e and any component H of G_e, H is dominated by $D^* \cap e$. In fact, any vertex $x \in V(H)$ is dominated by some vertex $y \in D^*$. Since $x \in C_e$, we have $y \in e$.

Set $\tilde{D}_e^* = D^* \cap e$. Suppose $\tilde{D}_e^*$ does not satisfy condition (ii). Then there is a component H of G_e such that H is not dominated by one connected component of $G[\tilde{D}_e^*]$. By Lemma 4, there are two components R and K of $G[\tilde{D}_e^*]$ such that $dist_H(R, K) \leq 3$. That is, R and K can be connected through at most two vertices in $V(H) \setminus \tilde{D}_e^*$. Add these vertices into $\tilde{D}_e^*$ to merge R and K. Continue this procedure until $\tilde{D}_e^*$ satisfies condition (ii). Suppose k mergences are executed. Then the resulting $\tilde{D}_e^*$ satisfies

$$|\tilde{D}_e^*| \leq |D^* \cap e| + 2k. \tag{4}$$

Next, we use vertices in $D_0(a^*) \cap e$ to compensate for the $2k$ term of inequality (4). Suppose the components are merged in the order that: H_1 is merged with H_2, H_3 is merged with H_4, ..., H_{2k-1} is merged with H_{2k}. To simplify the presentation of the idea, we first assume that the H_i's are all distinct components of the original $G[\tilde{D}_e^*]$. Denote by I_e the region of e between distance 1 and 2 from the boundary of e. For each $i = 1, 2, ..., k$, let x_i be a vertex in $V(H_{2i-1}) \cap I_e$. Such x_i exists since H_{2i-1} dominates some vertex in H which is a component in the central region of e (hence H_{2i-1} is within distance 1 from the central region), and $G[D^*]$ is connected (hence H_{2i-1} is accessible from the outer side of e. Because D_0 is a dominating set of G, there is a vertex $z_i \in D_0$ dominating x_i. Since $x_i \in I_e$, we have $z_i \in B_e$, and thus $z_i \in D_0(a^*) \cap e$. Note that for $i \neq j$, it is possible that $z_i = z_j$. However, in this case, x_i and x_j are independent since they are in different components of $G[\tilde{D}_e^*]$. Hence by Lemma 5, a vertex serves at most 12 times as z_i's. Thus we have shown that

$$k \leq 12|D_0(a^*) \cap e|. \tag{5}$$

Next, consider the case that there are some repetitions among the H_i's. For example, suppose H_3 is the component of the new $G[\tilde{D}_e^*]$ obtained by merging H_1 and H_2. Since x_1 is chosen to be in $V(H_1) \cap I_e$, we can choose $x_3 \in V(H_2) \cap I_e$.

In general, we are always able to choose x_i's such that they are in different components of the original $G[\tilde{D}_e^*]$. Hence (5) holds in any case. Combining (5) with (4), we have

$$|\tilde{D}_e^*| \leq |D^* \cap e| + 24|D_0(a^*) \cap e|. \tag{6}$$

Let $\tilde{D}$ be the union of the modified $\tilde{D}_e^*$'s, where e runs over all cells of $P(a^*)$. Then

$$|\tilde{D}| = \sum_{e \in P(a^*)} |\tilde{D}_e^*| \leq \sum_{e \in P(a^*)} (|D^* \cap e| + 24|D_0(a^*) \cap e|) = |D^*| + 24|D_0(a^*)|.$$

Hence $\tilde{D}$ satisfies requirements (i) and (ii). This completes the proof. □

4 Conclusion

We presented a construction and an analysis of PTAS for the minimum connected dominating set in unit ball graphs. This construction is different from that in [6] for the minimum connected dominating set in unit disk graphs. In fact, the construction in [6] cannot be extended to 3-dimensional space since a process of merging many parts of connected components into one in boundary area cannot work. Actually, our construction can be applied to unit ball graphs in n-dimensional space for any $n \geq 1$. In addition, when applied to unit disk graph, the $(1+\varepsilon)$-approximation constructed in this paper runs in time $n^{O(1/\varepsilon^2)}$ while the $(1+\varepsilon)$-approximation constructed in [6] runs in time $n^{O((1/\varepsilon^2)\ln(1/\varepsilon))}$. Therefore, Our construction also improves the running time.

References

1. Alzoubi, K.M., Wan, P., Frieder, O.: Message-optimal connected dominating sets in mobile ad hoc networks. In: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing, Lausanne, Switzerland, June 09-11 (2002)
2. Bharghavan, V., Das, B.: Routing in ad hoc networks using minimum connected dominating sets. In: International Conference on Communication, Montreal, Canada (June 1997)
3. Blum, J., Ding, M., Cheng, X.: Applications of Connectd Dominating Sets in Wireless Netwoks. In: Du, D.-Z., Pardalos, P. (eds.) Handbook of combinatorial Optimization, pp. 329–369. Kluwer Academic Publisher, Dordrecht (2004)
4. Butenko, S., Ursulenko, O.: On minimum connected dominating set problem in unit-ball graphs (submitted)
5. Cadei, M., Cheng, M.X., Cheng, X., Du, D.: Connected domination in ad hoc wireless networks. In: Proc. the Sixth International symposium on Mobile ad hoc networking and computing, Lausanne, Switzerland, June 09-11 (2002)
6. Cheng, X., Huang, X., Li, D., Wu, W., Du, D.: A polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks. Networks 42, 202–208 (2003)

7. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discrete Math. 86, 165–177 (1990)
8. Guha, S., Khuller, S.: Approximation algorithms for connected dominating sets. Algorithmica 20, 374–387 (1998)
9. Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. J. ACM 32, 130–136 (1985)
10. Min, M., Du, H., Jia, X., Huang, C.X., Huang, S.C., Wu, W.: Improving construction for connected dominating set with Steiner tree in wireless sensor networks. J. Global Optimization 35, 111–119 (2006)
11. Peter Chen, Y., Liestman, A.L.: Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks. In: Proceedings of the third ACM international symposium on Mobile ad hoc networking and computing, Lausanne, Switzerland, June 09-11 (2002)
12. Ramamurthy, B., Iness, J., Mukherjee, B.: Minimizing the number of optical amplifiers needed to support a multi-wavelength optical LAN/MAN. In: Proc. IEEE INFOCOM 1997, April 1997, pp. 261–268 (1997)
13. Ruan, L., Du, H., Jia, X., Wu, W., Li, Y., Ko, K.: A greedy approximation for minimum connected dominating set. Theoretical Computer Science 329, 325–330 (2004)
14. Salhieh, A., Weinmann, J., Kochha, M., Schwiebert, L.: Power Efficient topologies for wireless sensor networks. In: ICPP 2001, pp. 156–163 (2001)
15. Sivakumar, R., Das, B., Bharghavan, V.: An improved spine-based infrastructure for routing in ad hoc networks. In: IEEE Symposium on Computer and Communications, Athens, Greece (June 1998)
16. Stojmenovic, I., Seddigh, M., Zunic, J.: Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. In: Proc. IEEE Hawaii Int. Conf. on System Sciences (January 2001)
17. Wan, P., Alzoubi, K.M., Frieder, O.: Distributed construction of connected dominating set in wireless ad hoc networks. In: Proc. Infocom 2002 (2002)
18. Wu, J., Li, H.L.: On calculating connected dominating set for efficient routing in ad hoc wireless networks. In: Proceedings of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7–14 (1999)
19. Zong, C.: Shere pachings. Springer, New York (1999)