A Better Approximation Algorithm for Computing Connected Dominating Sets in Unit Ball Graphs

Donghyun Kim, *Student Member, IEEE*, Zhao Zhang, Xianyue Li, Wei Wang, Weili Wu, *Member, IEEE*, and Ding-Zhu Du, *Member, IEEE*

Abstract—A Virtual Backbone (VB) of a wireless network is a subset of nodes such that only VB nodes are responsible for routing-related tasks. Since a smaller VB causes less overhead, size is the primary quality factor of VB. Frequently, Unit Disk Graphs (UDGs) are used to model 2D homogeneous wireless networks, and the problem of finding minimum VBs in the networks is abstracted as Minimum Connected Dominating Set (MCDS) problem in UDGs. In some applications, the altitude of nodes can be hugely different and UDG cannot abstract the networks accurately. Then, Unit Ball Graph (UBG) can replace UDG. In this paper, we study how to construct quality CDSs in UBGs in distributed environments. We first give an improved upper bound of the number of independent nodes in a UBG, and use this result to analyze the Performance Ratio (PR) of our new centralized algorithm C-CDS-UBG, which computes CDSs in UBGs. Next, we propose a distributed algorithm D-CDS-UBG originated from C-CDS-UBG and analyze its message and time complexities. Our theoretical analysis shows that the PR of D-CDS-UBG is 14.937, which is better than current best, 22. Our simulations also show that D-CDS-UBG outperforms the competitor, on average.

Index Terms—Wireless networks, connected dominating sets, virtual backbones, unit ball graphs.

1 Introduction

Wireless networks including sensor networks and ad hoc networks consist of lots of wireless nodes, which are equipped with a processing unit, a communication module called a transceiver, a limited energy source such as a battery, etc., [1]. Because of the rapid improvements of VLSI, embedded sensor, and wireless radio communication technologies, the cost of wireless nodes is going down and their performance is getting better. Once wireless nodes are deployed on an application area, they organize an instant network autonomously without any predefined infrastructure. For these reasons, many people believe that wireless networks will play a key role in the next generation of networks. Currently, wireless networks are studied for various applications such as disaster rescue, environmental monitoring, battlefield surveillance, concert, health applications, and so on.

Radio signals, which are commonly used for wireless communications, consume energy that increases super-linearly proportional to their travel distance. In addition, each wireless node carries a limited power source. This makes energy efficiency a very important issue and multi-hop communication model is preferred in wireless networks. Since there is no predefined physical backbone infrastructure to support routing and topology control in wireless networks, routing-related tasks are very complicated and consume lots of energy. Furthermore, this situation is getting worse as the size of the network grows. To resolve the scalability problem in wireless networks and allow them to exploit the benefits of backbone infrastructure in wired networks, a backbone-like structure is introduced [2]. Nowadays, this is usually called a Virtual Backbone (VB).

Simply speaking, a VB of a wireless network is a subset of nodes in the network. Only nodes in VB are responsible for maintaining routing information and involved in routing tasks like machines in fixed physical backbones. Naturally, a smaller size VB is expected to suffer less from interference, generate less control messages, and to be more efficient. Therefore, size is a major quality factor for VBs in many previous works. So far, several methodologies are introduced to compute a quality VB in a wireless network. Among them, the Connected Dominating Set (CDS) problem is frequently used to model the problem of computing a minimum size VB in a wireless network, which was first attempted by Guha and Khuller [3] and later used in many works [4], [5], [6], [7], [8], [9], [10]. Since computing a Minimum CDS (MCDS) is a well-known NP-hard problem, all of the works attempted to generate approximated solutions.

In most cases, people studied the MCDS problem in homogeneous wireless networks. They also assumed that all nodes in the networks are on a two-dimensional space and used Unit Disk Graph (UDG) [12] to abstract the networks. However, sometimes, this assumption is very far

- *D. Kim, W. Wu, and D.-Z. Du are with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080. E-mail: donghyunkim@student.utdallas.edu, {weiliwu, dzdu}@utdallas.edu.*
- *Z. Zhang is with the College of Mathematics and System Sciences, Xinjiang University, Urumqi, Xinjiang 830046, P.R. China. E-mail: zhzhao@xju.edu.cn.*
- *X. Li is with the School of Mathematics and Statistics, Lanzhou University, Lanzhou, Gansu 730000, P.R. China. E-mail: lixianyue@lzu.edu.cn.*
- *W. Wang is with the Department of Mathematics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R. China. E-mail: wang_weiw@163.com.*

Manuscript received 2 Jan. 2009; revised 16 Oct. 2009; accepted 28 Feb. 2010; published online 14 Apr. 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-01-0001. Digital Object Identifier no. 10.1109/TMC.2010.55.

from reality. One good example is Underwater Sensor Networks (USNs) for ocean sampling networks, undersea explorations, distributed tactical surveillance, disaster prevention, assisted navigation, ocean environmental monitoring, etc., [13]. In many applications of USNs, underwater sensor nodes are floating around water instead of staying on the bottom of the ocean. In this case, the height of nodes becomes considerably important.

To abstract homogeneous wireless networks in three-dimensional space, Unit Ball Graphs (UBGs) model is frequently used [7], [14]. Since UDGs are special instances of UBGs in which the altitude of every node is the same, every NP-Hard problem in UDGs is also NP-Hard in UBGs. Naturally, MCDS in a UBG is still NP-Hard. To the best of our knowledge, there are three papers studying the problem of designing approximation algorithms to compute an MCDS of a UBG. Zou et al. proposed a centralized heuristic algorithm and proved that its Performance Ratio (PR) is $13 + \ln 10 \approx 15.302$ [15]. Butenko and Ursulenko introduced a distributed algorithm and showed that its PR is 22 [7]. Zhong et al. presented a distributed algorithm and claimed its approximation ratio to be 16 [14]. However, we found that Zhong et al.'s algorithm is flawed and corresponding PR analysis is incorrect. The details are discussed in the Appendix.

In this paper, we study the problem of constructing minimum CDSs in UBGs in distributed environments. The contributions of this paper can be summarized as follows:

1. We first figure out how many independent nodes can be contained in the union of two adjacent unit balls. We notice that this question is closely related to the famous Gregory-Newton Problem concerning with kissing number. We first transform our problem in two adjacent unit balls into another geometric problem in a graph embedded on the surface. Later, we use the Euler's Formula and plenty amount of calculus in the transformed problem and show that there can be at most 22 independent nodes inside the two adjacent unit balls. By combining our new result with some existing one, we improve the upper bound for the size of a Maximal Independent Set (MIS) in a UBG from $11\, OPT_{CDS} + 1$ in [7] to $10.917\, OPT_{CDS} + 1.083$, where OPT_{CDS} is the size of an optimal CDS of the UBG.
2. Instead of introducing our distributed algorithm directly, we first present a centralized algorithm C-CDS-UBG for which it is both simpler to understand its behavior and easier to show its PR than for the distributed algorithm. C-CDS-UBG computes an MIS first and connects nodes in the MIS using a simple greedy strategy to generate CDSs in UBGs. We use our improved upper bound on the size of an MIS in a UBG and prove that the PR of C-CDS-UBG is 14.937.
3. We introduce a distributed algorithm D-CDS-UBG, which is originated from C-CDS-UBG (i.e., same greedy strategy), and thus, whose PR is also 14.937. As a result, D-CDS-UBG is a distributed algorithm for the MCDS problem in UBGs, whose PR is better than the only and the best distributed algorithm for this problem in [7]. We also analyze its time and message complexities.
4. In simulations, we show that D-CDS-UBG outperforms the competitor's algorithm in [7], on average, which coincides with our theoretical analysis.

The rest of the paper is organized as follows: Section 2 discusses some related work. In Section 3, we derive an upper bound on the number of vertices in a maximal independent set in terms of the number of vertices in an optimal CDS. In Section 4, we first present C-CDS-UBG and prove its PR. Later, we introduce D-CDS-UBG and analyze its time and message complexities. In Section 5, we describe our simulation results and analyze them. At last, we make a conclusion and present some future work in Section 6.

2 Related Work

Around the end of 80s, the idea of VB for wireless networks was proposed by Ephremides et al. [2]. Guha and Khuller [3] introduced two polynomial-time two-stage greedy algorithms to compute an MCDS in a general graph G. The approximation ratio of the first one is $2(1 + H(\triangle))$, where $\triangle$ is the maximum degree of G and H is a harmonic function. In this algorithm, initially all nodes are colored white. Then, the algorithm selects a node with the maximum degree as a root and colors it black and its neighbors gray. Until there is no white node, the algorithm picks a set of white nodes S such that 1) each node in it is adjacent to a gray, 2) the number of adjacent white neighbors is the maximum, and 3) $1 \leq |S| \leq 2$, and colors the white nodes black and one of its gray neighbor black. After the algorithm is finished, the set of black nodes is a CDS. The second algorithm is a two-phase algorithm. In the first phase, it computes a dominating set. Later, it connects nodes in the set using a Steiner tree algorithm. In this way, it achieves the PR of $H(\triangle) + 2$. Later, Ruan et al. modify this algorithm to a single-stage algorithm whose PR is $(\ln(\triangle) + 2)$ [6].

Wan et al. introduced a two-stage distributed CDS computation algorithm whose approximation ratio is a constant 8 [4]. To model a wireless network, it uses a UDG. In the first stage of this algorithm, it computes an MIS, which is also a dominating set. In the next stage, the nodes in the MIS are connected using a Steiner tree algorithm. Cardei et al. [5] also introduced a two-phase distributed algorithm. The PR of this algorithm is also 8, but its message complexity is lower. Recently, Wu et al. improved the lower bound of MISs in UDGs from $4opt + 1$ to $3.8opt + 1$ using a geometrical analysis, where opt is an optimal CDS in a UDG [8]. Using this result, the PR of both [4] and [5] is improved from 8 to 7.8. In [10], Funke et al. improved the lower bound to $3.45opt + 1$ and used this result to provide a 6.91-approximation algorithm to compute CDSs in UDGs. Very recently, Li et al. further improved the bound to $3.4306opt + 4.8185$ and provided a 6.075-approximation algorithm to compute CDSs in UDGs [11].

In most cases, people assume that wireless nodes are on a two-dimensional plane, and to model a homogeneous wireless network, they use a UDG. However, in reality, the altitude of nodes can be very different from each other (i.e., USNs [13]). Recently, people are using UBGs to model homogeneous wireless networks in three-dimensional space. In a UBG, vertices are representing nodes in a wireless network and there is an edge between two vertices if their distance is smaller than the maximum transmission range of the nodes. Hansen and Schmutz studied the expected size of a CDS in a random UBG [17]. Very recently, Butenko and Ursulenko introduced a distributed algorithm to compute an MCDS in a UBG and proved that the PR of their algorithm is 22 [7]. Zou et al. proposed a centralized algorithm whose PR is $13 + \ln 10 \approx 15.302$ [15].

3 An Improved Upper Bound for Maximal Independent Sets

This section introduces a better upper bound for the number of vertices in an MIS in terms of the number of vertices in an optimal CDS. In the following sections, we prove the following things. First, we consider how many independent nodes can be contained in a unit ball. Second, we establish an upper bound for the number of independent nodes in two adjacent unit balls. At last, based on the results from the previous two sections, we obtain the main result of this section.

Here are some notations and definitions, which are frequently used in the rest of this section. For two points P and Q in a three-dimensional space, $d(P,Q)$ denotes the euclidean distance between them. For a geometric object B, denote the outer surface of B by $Sur(B)$. $N_G(v)$ denotes the neighborhood of v in G. We say that $N_G(u)$ and $N_G(u')$ are adjacent if $d(u,u') \leq 1$. Otherwise, they are independent. Likewise, any two nodes v and v' are adjacent if $d(v,v') \leq 1$. Otherwise, they are independent. At last, we say that two unit balls are touching each other if they have exactly one point in common, and the common point is called *touching point*.

3.1 An Upper Bound on the Number of Independent Nodes in a Unit Ball

In this section, we establish an upper bound on the number of independent nodes in a unit ball. This can be done by transforming this question into the famous Gregory-Newton Problem concerning about kissing number. The *kissing number* $k(S_3)$ (see [16] for reference) is the maximum number of independent unit balls that can simultaneously touch the surface of one unit ball. Newton conjectured in 1694 that $k(S_3) = 12$. The conjecture was not confirmed until 1874 by Hoppe [18].

Lemma 3.1. *The kissing number* $k(S_3) = 12$.

As a corollary, we have

Corollary 3.2 ([19]). $\Delta(T)$, *the maximum degree of a minimum spanning tree* T *in a UBG* G *is at most 12.*

In addition, we present the following useful lemma:

Lemma 3.3 ([19]). *For any vertex* u *in a UBG* G, *the neighborhood* $N_G(u)$ *contains at most 12 independent vertices.*

3.2 An Upper Bound on the Number of Independent Nodes in Two Adjacent Unit Balls

In this section, we focus on the problem asking how many independent nodes can be contained in two adjacent unit balls. As a consequence of Lemma 3.3, there are at most 24 independent nodes in the union of two unit balls. In this section, we use Euler's Formula to improve this upper bound to 22.

Let B_1 and B_2 be two adjacent unit balls with centers u_1 and u_2, respectively. Suppose that $I = \{v_1, v_2, \ldots, v_t\}$ is the set of independent nodes contained in $(B_1 \cup B_2) \setminus (B_1 \cap B_2)$. In order to obtain the main result of this section, we shall show that $t \leq 20$.

The basic idea of this proof is similar to that in deriving an upper bound for kissing number in [16]. First, we "project" each independent node in I to the surface of $B_1 \cup B_2$. Then, join the points with curves on the surface $Sur(B_1 \cup B_2)$ to divide $Sur(B_1 \cup B_2)$ into patches, and thus, obtain a graph *embedding* on the sphere (each patch is a *face* in terms of Graph Theory). By properly choosing the way of drawing the curves, lower bounds for the areas of faces can be obtained. Finally, using the fact that the sum of the areas of all faces is the area of $Sur(B_1 \cup B_2)$ and applying Euler's Formula $t + f - e = 2$, where t, f, e are the number of vertices, faces, and edges in the embedded graph, respectively, the upper bound for the number of vertices can be obtained.

For this purpose, we first "project" independent nodes in I to the surface of $B_1 \cup B_2$ as follows: For each $i = 1, 2, \ldots, t$, if $v_i \in B_1 \setminus B_2$ (respectively, $B_2 \setminus B_1$), we draw a radial from u_1 (respectively, u_2) going through v_i, and denote by P_i the intersection point of this radial with $Sur(B_1)$ (respectively, $Sur(B_2)$). Then, $\{P_1, P_2, \ldots, P_t\}$ are all on $Sur(B_1 \cup B_2)$. Furthermore, since v_is are independent, we see that for any two distinct integers i and j, $d(P_i, P_j) > 1$, that is, P_is are also independent.

There are two difficulties, which are different from dealing with kissing number in one ball. The first is how to draw the curves to connect the points, especially when two points are in different balls. The second is how to obtain the lower bounds for the areas of the faces. In fact, it is easier to deal with faces lying completely on the surface of one ball (call such a face *regular*). When a face "strides over" two balls (call such a face *striding*), the area is much smaller than that of a regular one. Hence, we have to deal with these two different types of faces separately and find out an upper bound for the number of striding faces.

3.2.1 Partition of the Surfaces of $B_1 \cup B_2$

In this section, we introduce a way to draw curves connecting P_is. We are going to establish a curve between any P_i and P_j whose distance is between 1 and 3 arccos(1/7)π. Then, by [16], such drawing will give us a planar graph embedded on the union of the surfaces $Sur(B_1)$ and $Sur(B_2)$. Later, we will use this graph to estimate the maximum number of independent nodes in two adjacent unit balls. Clearly, the intersection of $Sur(B_1)$ and $Sur(B_2)$ is a circle. Let us denote it by L. Now, for any two points P_i and P_j such that $1 < d(P_i, P_j) < 3\arccos(1/7)/\pi$, join them by a curve on $Sur(B_1 \cup B_2)$ in the following way:

Case 1. If they are on the surface of a same unit ball, join them by a geodesic arc ℓ. If ℓ does not completely lie on $Sur(B_1 \cup B_2)$, then ℓ intersects L at two points Q_1 and Q_2 (see Fig. 1). Without loss of generality, suppose that Q_1 is nearer to P_i on arc ℓ. The curve joining P_i and P_j is composed of the union of the three arcs $\ell_{P_iQ_1}$, $L_{Q_1Q_2}$, and $\ell_{Q_2P_j}$, where $\ell_{P_iQ_1}$ is the segment on arc ℓ between the two points P_i and Q_1, etc.

Case 2. If both P_i and P_j are on the surfaces of different unit balls, say, $P_i \in Sur(B_1)$ and $P_j \in Sur(B_2)$, let Π be the plane going through the three points P_i, P_j, and Q_0, where

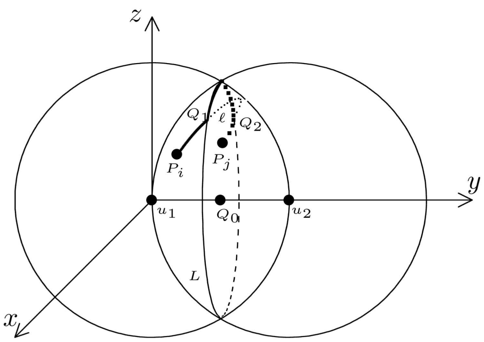


Fig. 1. Join two points on the surface of a same ball.

Q_0 is the middle point of the line segment u_1u_2 (see Fig. 2). Then Π intersects with the circle L at two points, let Q be the one which is nearer to P_i and P_j. Join P_i and P_j to Q by geodesic arcs ℓ_{P_iQ} and ℓ_{QP_j}. The curve joining P_i and P_j is composed of the union of ℓ_{P_iQ} and ℓ_{QP_j}.

Note that no two curves drawn can intersect. This is because, on one hand, since the distance between any two P_is is greater than 1, it can be calculated that for any four points, the two diagonals of the quadrilateral formed by the four points cannot be both smaller than $3\arccos(1/7)/\pi$. On the other hand, by our construction, only those points at distance smaller than $3\arccos(1/7)/\pi$ are joined. Hence, a curve can meet another curve only at same point P_i, and thus, gives a sphere embedding of a graph. Recall that the Euler's Formula can only be applied to a concrete embedding of a graph.

3.2.2 Lower Bound for the Area of the Face

Now we give the lower bounds for the areas of the faces of the above concrete embedding of graph. For a face S bounded by k-curves, we call it a *k-face*. Denote by $A(S)$ the area of S. Let A_k be the minimum area of a regular k-face and $\widetilde{A}_k$ be the minimum area of a striding k-face. A k-face whose boundary is "cut" by L as in Case 1 is regarded as a striding face. A k-face whose boundary "touch" L at one point is regarded as regular.

Lemma 3.4. *For the regular faces of the above concrete embedding of graph,* $A_3 = 0.5512\ldots$, $A_4 = 1.3338\ldots$, $A_5 = 2.2261\ldots$. *In general,*

$$A_k \geq (k-2)A_3. \tag{1}$$

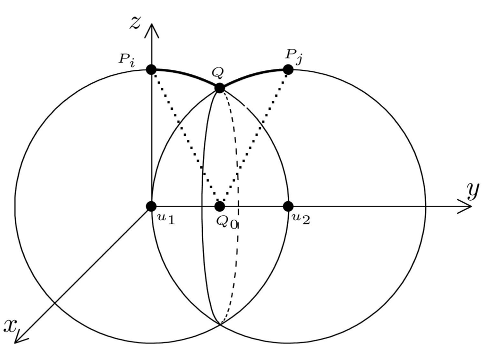


Fig. 2. Join two points on the surfaces of different balls.

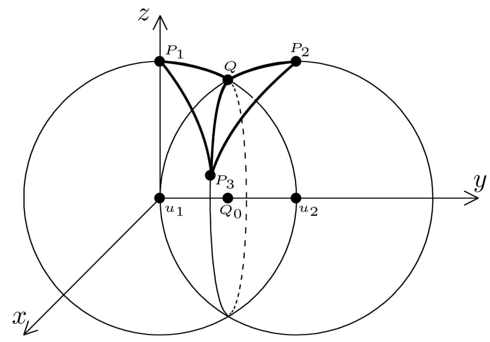


Fig. 3. An extreme position for $\widetilde{A}_3$. The coordinates of P_1, P_2, P_3 are $P_1 = (0,0,1)$, $P_2 = (0,1,1)$, and $P_3 = (\sqrt{2}/2, 1/2, 1/2)$ (note that $d(P_1,P_2) = d(P_2,P_3) = d(P_3,P_1) = 1$).

For the striding faces, $\widetilde{A}_3 = 0.4076\ldots$, $\widetilde{A}_4 = 0.9949\ldots$, $\widetilde{A}_5 = 1.8732\ldots$. *In general,*

$$\widetilde{A}_k \geq (k-2)\widetilde{A}_3 \quad \text{for } k = 3,4,5,\ldots. \tag{2}$$

Proof. The values for A_ks can be found in [16, Page 11], and the positions for k-faces whose areas achieve these extreme values can also be found there. For a striding k-face, it can be calculated by integral using polar coordinate that $\widetilde{A}_3 = 0.4076\ldots$, and $\widetilde{A}_3$ is achieved if the position of the 3-face is like the one shown in Fig. 3; $\widetilde{A}_4 = 0.9949\ldots$, and $\widetilde{A}_4$ is achieved if the position of the 4-face is like the one in Fig. 4; $\widetilde{A}_5 = 1.8732\ldots$, and $\widetilde{A}_5$ is achieved if the position of the 5-face is similar to that in Fig. 5, etc. By using a similar approach used in [16] for A_ks, we can prove that

$$\widetilde{A}_k \geq (k-2)\widetilde{A}_3 \quad \text{for } k = 3,4,5,\ldots. \tag{3}$$

□

3.2.3 Main Results

Lemma 3.5. *The number of independent nodes in* $(B_1 \cup B_2) \setminus (B_1 \cap B_2)$ *is at most 20.*

Proof. The proof of this lemma is equivalent to show $t \leq 20$. Denote by f_i the number of all i-faces (including regular i-faces and striding i-faces), and $\widetilde{f}_i$ the number of striding i-faces that strides over L.

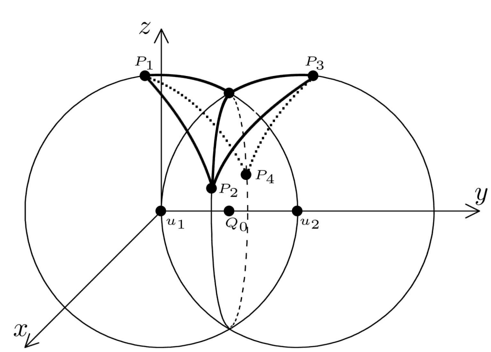


Fig. 4. An extreme position for $\widetilde{A}_4$. The coordinates of the points are $P_1 = (0, -0.1816, 0.9834)$, $P_2 = (0.6241, 0.5, 0.6004)$, $P_3 = (0, 1.1816, 0.9834)$, and $P_4 = (-0.6241, 0.5, 0.6004)$ (note that $d(P_1,P_2) = d(P_2,P_3) = d(P_3,P_4) = d(P_4,P_1) = 1$ and $d(P_1,P_3) = 3\arccos(1/7)/\pi$).

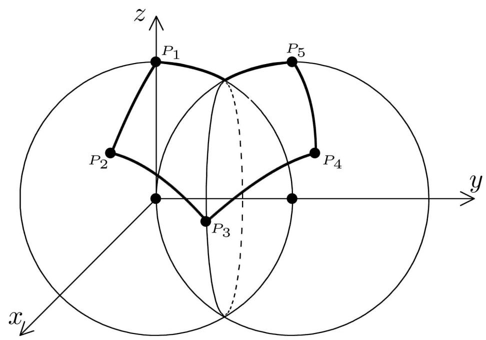


Fig. 5. An extreme position for $\widetilde{A}_5$. The coordinates of the points are $P_1 = (0, 0, 1)$, $P_2 = (0.7690, -0.3984, 0.5)$, $P_3 = (0.8631, 0.5, 0.0710)$, $P_4 = (0.7690, 1.3984, 0.5)$, and $P_5 = (0, 1, 1)$ (note that $d(P_1, P_2) = d(P_2, P_3) = d(P_3, P_4) = d(P_4, P_5) = d(P_5, P_1) = 1$ and $d(P_3, P_1) = d(P_3, P_5) = 3\arccos(1/7)/\pi$).

By Euler's Formula and Handshaking Theorem [20],

$$\begin{aligned} 2t - 4 &= 2e - 2f \\ &= 3f_3 + 4f_4 + 5f_5 + \cdots - 2(f_3 + f_4 + f_5 + \cdots) \\ &= f_3 + 2f_4 + 3f_5 + \cdots . \end{aligned} \tag{4}$$

Then, by Lemma 3.4 and (4), we have

$$\begin{aligned} 6\pi &\geq A_3(f_3 - \widetilde{f}_3) + \widetilde{A}_3\widetilde{f}_3 + A_4(f_4 - \widetilde{f}_4) + \widetilde{A}_4\widetilde{f}_4 \\ &\quad + A_5(f_5 - \widetilde{f}_5) + \widetilde{A}_5\widetilde{f}_5 + \cdots \\ &\geq A_3(f_3 + 2f_4 + 3f_5 + \cdots) \\ &\quad - (A_3 - \widetilde{A}_3)(\widetilde{f}_3 + 2\widetilde{f}_4 + 3\widetilde{f}_5 + \cdots) \\ &= A_3(2t - 4) - (A_3 - \widetilde{A}_3)(\widetilde{f}_3 + 2\widetilde{f}_4 + \cdots), \end{aligned} \tag{5}$$

where 6π is an upper bound for the area of $Sur(B_1 \cup B_2)$. This equation means that if we find the upper bound of $\widetilde{f}_3 + 2\widetilde{f}_4 + \cdots$, we can get the maximum value of t.

We first assume that every striding face is a 3-face. Let $S_1 = P_1P_2P_3$ and $S_2 = P_2P_3P_4$ be two 3-faces such that the common arc P_2P_3 goes through L (see Figs. 6a and 6b). Then the projection of $S_1 \cup S_2$ onto L has arc length at least 0.8744, which can be obtained from the extreme position shown in Fig. 6c. Since the arc length of L is $\sqrt{3}\pi$, and $\sqrt{3}\pi/0.8744 = 6.223$, we see that there are at most six such pairs of 3-faces. Hence, $\widetilde{f}_3 \leq 13$. Recall that we have assumed that $\widetilde{f}_i = 0$ for $i > 3$ and have proved that $\widetilde{f}_3 \leq 13$. Hence, it follows from (5) that $t \leq 20.792$. Since t is an integer, we have $t \leq 20$.

Next, suppose that there are striding i-faces for some $i > 3$. We shall transform this situation into one which can be dealt with by the above analysis. This is done by further dividing striding i-faces ($i > 3$) into smaller ones satisfying the following three properties that are essential to the above analysis:

1. Each striding 3-face has area at least $\widetilde{A}_3$.
2. Each regular k-face has area at least $(k-2)A_3$.
3. The projection of each pair of striding 3-faces onto L occupies arc length at least 0.8744 (hence, $\widetilde{f}_3$ cannot be too large).

Let $S = P_1P_2P_3P_4$ be a striding 4-face. If each ball contains two of P_is, say, $P_1, P_2 \in Sur(B_1)$ and $P_3, P_4 \in Sur(B_2)$ as in Fig. 7a, then we can join P_2 with P_3 with a curve such that each of the two 3-faces $P_1P_2P_3$ and $P_2P_3P_4$ has area at least $\widetilde{A}_3$ (this can be done since $\widetilde{A}_4 \geq 2\widetilde{A}_3$). Moreover, S can be viewed as the union of a pair of striding 3-faces such that the projection of them onto L occupies arc length greater than 0.8744.

Now, suppose that one ball contains only one of P_is, say, $P_1, P_2, P_3 \in Sur(B_1)$ and $P_4 \in Sur(B_2)$ (see Fig. 7b). Join P_2 with P_3 with a curve such that $A(P_1P_2P_3) = 0.5512$. Recall that $A(S) \geq 0.9949$, we have $A(P_2P_3P_4) \geq 0.4437 > \widetilde{A}_3$. Hence, S can be viewed as the union of a regular 3-face and a striding 3-face. Furthermore, the projection of the striding 3-face $P_2P_3P_4$ onto L occupies arc length no smaller than that of an original striding 3-face does. This is because P_2 and P_3 are not joined in our construction, which means $d(P_2, P_3) \geq 3\arccos(1/7)/\pi >$ the euclidean distance between the end points of a side of an original striding 3-face.

Let $S = P_1P_2P_3P_4P_5$ be a striding 5-face. If one ball contains two of P_is, say, $P_1, P_2, P_3 \in Sur(B_1)$ and $P_4, P_5 \in Sur(B_2)$ (see Fig. 8a). Join P_5 with P_2 and P_3 such that each of the three new 3-faces $P_1P_2P_5$, $P_2P_3P_5$, $P_3P_4P_5$ has areas greater than $\widetilde{A}_3$ (this can be done since $\widetilde{A}_5 \geq 3\widetilde{A}_3$). If one ball contains only one of P_is, say, $P_5 \in Sur(B_2)$ and $P_1, \ldots, P_4 \in Sur(B_1)$ (see Fig. 8b), join P_1 with P_4 with a curve such that $A(P_1P_2P_3P_4) = 1.3338$. Since $A(S) \geq 1.8732$, we have $A(P_1P_4P_5) \geq 0.5349 > \widetilde{A}_3$.

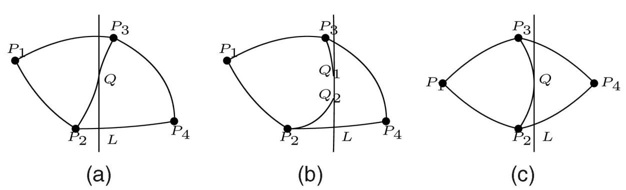


Fig. 6. A pair of 3-faces striding over L. (c) The extreme position that the projection of $S_1 \cup S_2$ onto L has arc length 0.8744, where $d(P_1, P_2) = d(P_1, P_3) = d(P_2, P_4) = d(P_3, P_4) = d(P_2, P_3) = 1$.

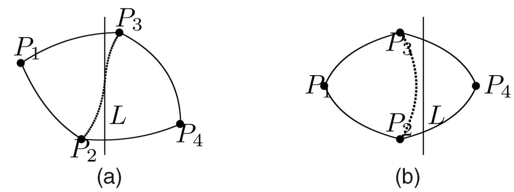


Fig. 7. Dividing a 4-gon striding over L.

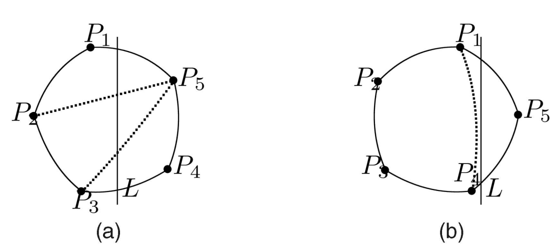


Fig. 8. Dividing a 5-gon striding over L.

In general, for any i-face with $i > 3$, we can divide it into some regular j-faces with area at least A_j and some striding 3-faces with area at least $\widetilde{A}_3$. Furthermore, the projection of a new striding 3-face thus obtained onto L occupies an arc length no smaller than that of an original striding 3-face does. Hence, by a similar argument as before, we have $t \leq 20$. □

Now, we are ready to prove our main lemma in this section.

Lemma 3.6. *The number of independent nodes in the union of two adjacent unit balls is at most 22.*

Proof. Let B_1, B_2 be two adjacent unit balls. Suppose that $I = \{v_1, \dots, v_s\}$ is the set of independent nodes in $B_1 \cup B_2$. If $B_1 \cap B_2$ contains at least two of v_is, then, by Lemma 3.3, $s \leq 2 \times 12 - 2 = 22$, and we are done. Hence, suppose that $B_1 \cap B_2$ contains at most one of v_is. According to Lemma 3.5, $(B_1 \cup B_2) \setminus (B_1 \cap B_2)$ contains at most 20 of v_is and $s \leq 20 + 1 = 21$. □

Then, following the same line of [8, Theorem 1], the main result of this section can be obtained.

Theorem 3.7. *Let M be an MIS of a UBG G. Then $|M| \leq 10.917\ OPT_{CDS} + 1.083$, where OPT_{CDS} is the number of vertices in an optimal CDS of G.*

Proof. Let C be a connected dominating set of G with $|C| = OPT_{CDS}$ and $G[C]$ be the subgraph of G induced by C. Note that $G[C]$ is a UBG since G is a UBG. By Corollary 3.2, $G[C]$ has a minimum spanning tree T with maximum degree at most 12. We will show that there are at most $10.917|T| + 1.083$ independent nodes in the neighborhood of T by induction on $|T|$. When $|T| = 1$ or 2, the assertion is true by Lemmas 3.3 and 3.6. Hence, we suppose that $|T| \geq 3$. Since T is a tree, there is a nonleaf node v such that it is adjacent to at most one nonleaf node.

Let u be the nonleaf neighbor of v or the root of T. Now, suppose that $x_1, \dots, x_k (k \leq 11)$ is a set of leaf neighbors of v. Then, $T' = T - \{v, x_1, \dots, x_k\}$ is a minimum spanning tree of $G[C - \{v, x_1, \dots, x_k\}]$, which is a UBG induced by all the nodes in $C - \{v, x_1, \dots, x_k\}$. Furthermore, the maximum degree of T' is 12 due to the following reasons: First, since T is a minimum spanning tree with maximum degree 12, T' is also a spanning tree of $G[C - \{v, x_1, \dots, x_k\}]$ with maximum degree 12. Now, suppose that T' is not a minimum spanning tree. Then, there has to be another minimum spanning tree T''. Now, $T'' + \{uv, vx_1, \dots, vx_k\}$ is a spanning tree of $G[C]$ with a smaller weight than T, which contradicts to our assumption that T is a minimum spanning tree.

By inductive hypothesis, there are at most $10.917(|T| - k - 1) + 1.083$ independent nodes in the neighborhood of $T - \{v, x_1, \dots, x_k\}$. Furthermore, by Lemma 3.3, for any node $x_i (1 \leq i \leq k)$, there are at most 11 independent nodes in its neighborhood also independent from v and by Lemma 3.6, the neighborhood of v and x_k contains at most 21 independent nodes also independent from u. Therefore, there are at most

$$
\begin{aligned}
&10.917(|T| - k - 1) + 1.083 + 21 + 11(k - 1) \\
&\quad = 10.917|T| + 1.083 + 0.083k - 0.917 \\
&\quad \leq 10.917|T| + 1.083
\end{aligned}
$$

independent nodes in the neighborhood of T. □

4 CDS-UBG: A New Connected Dominating Set Computation Algorithm in Unit Ball Graphs

So far, we have shown that the maximum number of isolated nodes in a unit ball is no more than 22. In this section, we introduce a distributed greedy algorithm, D-CDS-UBG, for the MCDS problem in UBGs. Before introducing D-CDS-UBG, we first present its centralized version, C-CDS-UBG. This is because this centralized version is both simpler to understand its behavior and easier to show its PR than D-CDS-UBG. Note that C-CDS-UBG is a round-based algorithm and each round requires to test a small number of nodes, which are adjacent to processed nodes in the previous rounds. Therefore, C-CDS-UBG is easy to be converted into a distributed algorithm. We show that the PR of C-CDS-UBG is 14.937, and so is that of D-CDS-UBG's by exploiting our results in the previous section. We also prove that both the time and message complexities of D-CDS-UBG is $O(n^2)$. The following notations are frequently used in the this section:

- $G = (V, E) = (V(G), E(G))$ is a connected UBG.
- $Eucdist(u, v)$ is the euclidean distance between two nodes u and v in $V(G)$.
- $N(u) = \{v | v \in V(G) \setminus \{u\} \text{ and } Eucdist(u, v) \leq 1\}$.
- $N[x] = N(x) \cup \{x\}$.
- For a node set C, $N(C) = (\bigcup_{x \in C} N(x)) \setminus C$.
- $M_{v,C}$ is a set of MIS nodes adjacent with v and not in C.

4.1 C-CDS-UBG: A Centralized CDS Computation Algorithm in UBGs

C-CDS-UBG is formally described in Algorithm 1. Roughly, this algorithm works as follows: Given $G = (V, E)$, it first computes an MIS M_0 such that for any $M \subset M_0$, the distance between M and $M_0 \setminus M$ is exactly two hops. To compute such M_0, we use the idea in [7]. To generate a CDS, the algorithm connects nodes in M_0 by selecting some nodes in $V \setminus M_0$ using a greedy strategy. C-CDS-UBG is a round-based algorithm. It puts one MIS node in a set C and grows C by repeatedly adding a non-MIS node v together with its adjacent MIS nodes such that $|M_{v,C}|$ is maximized. Note that the size of the tested node in each round is small and they are adjacent to C. Therefore, it is easy to build a distributed algorithm from C-CDS-UBG. Now, we introduce some lemmas and theorems to prove the PR of Algorithm 1.

Algorithm 1. C-CDS-UBG ($G(V, E)$)

1: Set $M_0 = \emptyset, W = \emptyset, V' = V$.
2: Pick a root $r \in V'$ with a maximum degree.
3: Set $M_0 = \{r\}$, $W = N(r)$, and $V' = V' \setminus (\{r\} \cup N(r))$.
4: **while** $V' \neq \emptyset$ **do**
5:     Pick $x \in N(W)$ such that $|N(x) \cap V'|$ is maximized.

6: Set $M_0 = M_0 \cup \{x\}$, $W = W \cup (N(x) \cap V')$, and $V' = V' \setminus (\{x\} \cup N(x))$.
7: **end while**
8: Set $C = \{r\}$ and $M = M_0 - \{r\}$.
9: **while** $M \neq \emptyset$ **do**
10: Pick a node $v \in N(C)$ such that $|M_{v,C}| = \max\{|M_{x,C}| \mid x \in N(C)\}$.
11: Set $C = C \cup \{v\} \cup M_{v,C}$ and $M = M \setminus M_{v,C}$.
12: **end while**
13: Return C.

Lemma 4.1. *Suppose that M_0 is obtained after Line 7 of C-CDS-UBG. Then, for any $M \subset M_0$, the distance between M and $M_0 \setminus M$ is exactly two hops.*

Proof. In Algorithm 1, W is a set of nodes, which are adjacent to M_0 and after Line 3, $M_0 = \{r\}$. In Line 5, to grow M_0, we pick $x \in N(W)$. Since x is adjacent to a node in W, which is adjacent to a node in M_0, the hop distance between x and M_0 is exactly two hops. In Line 6, we set $M_0 = M_0 \cup \{x\}$, and thus, for any $M \subset M_0$, the distance between M and $M_0 \setminus M$ becomes exactly two hops. □

Theorem 4.2. *After Algorithm 1 is executed, M_0 is an MIS and C is a CDS.*

Proof. By its definition, M_0 is an MIS if

1. $\forall v \in V$, either $v \in M_0$ or $\exists u \in M_0 : v \in N(u)$ and
2. $\forall u, v \in M_0, (u, v) \notin E(G)$.

In Lines $3 \sim 7$, when a node in $N(W)$ is included in M_0, its neighbors are included in W, and thus, those neighbors cannot be included in M_0 later, which means that there cannot be two adjacent MIS nodes. Furthermore, W is a set of nodes, which are adjacent at least one node in M_0 and after Line 7, all nodes are in M_0 or W since V' is empty. Therefore, M_0 is an MIS.

After Line 8, C contains only one node r. Since the distance between any two nodes in M_0 is exactly two hops, as long as C is an incomplete CDS, there has to be a node $v \in N(C)$ such that v is connecting C and some nodes in M. And therefore, by selecting such v repeatedly, we can connect r to the rest of other nodes in M_0. Since M_0 is an MIS and we connect them, C is a CDS. □

Lemma 4.3. *For any node $u \in G$, $N(u)$ contains at most 12 independent nodes.*

Proof. Since G is a UBG, this is true by Lemma 3.3. □

Lemma 4.4. *After Algorithm 1 is finished, $|C \setminus M_0| \leq 4.02|OPT_{CDS}|$, where OPT_{CDS} is an optimal CDS of G.*

Proof. Suppose $OPT_{CDS} = \{x_1, x_2, \ldots, x_t\}$, which is an optimal CDS of G. Let

- $S_i = \{x_i\} \cup (N(x_i) \setminus OPT_{CDS})$ for $i = 1$, and
- $S_i = \{x_i\} \cup ((N(x_i) \setminus OPT_{CDS}) \setminus \bigcup_{j=1}^{i-1} S_j)$ for $i \leq t$.

Then, $\{S_1, S_2, \ldots, S_t\}$ forms a partition of $V(G)$.

Let's consider the following weighting scheme. That is, when we select a node v and add $\{v\} \cup M_{v,C}$ to C, each MIS node x in $M_{v,C}$ is assigned a weight $w(x) = 1/|M_{v,C}|$. Note that if C is not a CDS, then, by Lemma 4.1, we have $|M_{v,C}| > 0$. Also,

$$\sum_{\forall S_i} \sum_{x \in M_0 \cap S_i} w(x)$$

is the upper bound of $|C \setminus M_0|$ in Algorithm 1.

Now, we show that for each i, $\sum_{x \in M_0 \cap S_i} w(x) \leq 4.02$. When $M_0 \cap S_i = \emptyset$, the proof is trivial, and thus, we assume that $M_0 \cap S_i \neq \emptyset$. Denote by $a_j = (M_0 \cap S_i) \setminus C_j$, where C_j is the set C after the jth iteration, and $C_0 = \{r\}$ is the initial set. Suppose that j_e is the first index of the iteration after which all nodes of $M_0 \cap S_i$ are included in C. Hence, $a_j > 0$ for $j = 0, 1, \ldots, j_e - 1$ and $a_{j_e} = 0$. To simplify our statement, we assume $a_{j-1} - a_j > 0$ for $j = 1, 2, \ldots, j_e$. After the first iteration, the number of MIS nodes which receive weights is $a_0 - a_1$ and the weight assigned to each node is at most $1/(a_0 - a_1)$. Since $a_0 - a_1 > 0$, some MIS nodes in $M_0 \cap S_i$ are added to C in the first iteration. Hence, before the jth iteration for $j \in \{2, 3, \ldots, j_e\}$, x_i is adjacent to C. By the greedy strategy, we see that $|M_{v_j,C_{j-1}}| \geq |M_{x_i,C_{j-1}}|$, where v_j is the non-MIS node chosen in the jth iteration by the algorithm. After the jth iteration, $a_{j-1} - a_j$ nodes of $M_0 \cap S_i$ receive weights and the weight assigned to each node is $1/|M_{v_j,C_{j-1}}| \leq 1/|M_{x_i,C_{j-1}}| = 1/a_{j-1}$. Therefore,

$$\sum_{x \in M_0 \cap S_i} w(x) \leq \frac{1}{a_0 - a_1}(a_0 - a_1) + \sum_{j=2}^{j_e} \frac{1}{a_{j-1}}(a_{j-1} - a_j).$$

Since we have assumed that $a_{j-1} - a_j > 0$ for any $j = 1, 2, \ldots, j_e$, the number of MIS nodes in $(M_0 \cap S_i) \setminus C_{j-1}$ strictly decreases in every iteration. Hence, by Lemma 4.3, $j_e \leq 12$ and the second term on the right of the above expression is bounded by $H(11)$, where H is the harmonic function. Therefore, we have

$$\sum_{x \in M_0 \cap S_i} w(x) \leq 1 + H(11) \leq 4.02.$$

□

Theorem 4.5. *The size of the node set C generated by Algorithm 1 is no more than $14.937|OPT_{CDS}| + 1.083$, where OPT_{CDS} is an optimal CDS of G and $|OPT_{CDS}| = t$.*

Proof. By Theorem 3.7, $|M_0| \leq 10.917|OPT_{CDS}| + 1.083$, and by Lemma 4.4, Algorithm 1 adds at most $4.02|OPT_{CDS}|$ to connect the nodes in M_0. Therefore,

$$\begin{aligned} |C| &= \sum_{i=1}^{t} \sum_{x \in M_0 \cap S_i} w(x) + |M_0| \\ &\leq 4.02t + 10.917|OPT_{CDS}| + 1.083 \\ &\leq 14.937|OPT_{CDS}| + 1.083. \end{aligned}$$

□

4.2 D-CDS-UBG: A Distributed CDS Computation Algorithm in UBGs

In this section, we introduce D-CDS-UBG, a distributed version of C-CDS-UBG. To compute a CDS of a UBG, D-CDS-UBG first runs D-MIS-UBG to get an MIS in a distributed manner. Then, it connects the MIS nodes by adding some nodes so that a subgraph induced by them is connected. We skip the PR analysis of D-CDS-UBG since its greedy strategy is equal to C-CDS-UBG's, and so are their PRs.

4.2.1 D-MIS-UBG: A Distributed MIS Computation Algorithm in UBGs

D-MIS-UBG is a simple coloring algorithm to compute an MIS of a UBG. This algorithm assumes that all nodes in a given graph are initially colored white. As the algorithm proceeds, a node is colored black if it is chosen to be in an MIS. Every node adjacent to a black node is colored gray. After this algorithm is terminated, the set of black nodes is an MIS. Now, we present the description of D-MIS-UBG. Note that in the description of D-MIS-UBG, $C(x)$ denotes a set of children of x, and $P(x)$ denotes a parent of x.

1. For every node x, $C(x) = N(x)$. Select a node r with the maximum degree as the root. r colors itself black and the nodes in $C(r)$ gray. Every node $x \in C(r)$ sets $P(x) = r$ and $C(x) = C(x) \setminus N[r]$.
2. r broadcasts an ASK message to white nodes $x \in C(y)$ for all gray nodes y. When a nonwhite node receives this message, it forwards this to its children nodes. When a white node x receives the ASK message, it sends back an $REP(x, m(x))$ message to r, where $m(x)$ is the number of white nodes in $N[x]$.
3. When a nonwhite node x receives REP messages from all nodes in $C(x)$, it picks a node $y \in C(x)$ such that $m(y)$ is maximum, sets $Index(x) = y$, $m(x) = m(y)$, and sends $REP(x, m(x))$ message to $P(x)$.
4. When the root r receives REP messages from all nodes in $C(r)$, it picks one node $x \in C(r)$ such that $m(x)$ is maximum. If $m(x) = 0$, then the set of black nodes is an MIS and r finishes D-MIS-UBG. Otherwise, r sends a $JOIN$ message to x.
5. On receiving a $JOIN$ message by a node x, if x is gray, then it forwards the $JOIN$ message to the node $Index(x)$. If x is white, then x colors itself black and all white nodes in $C(x)$ gray, $P(x)$ is set to be the node which sends $JOIN$ message to x, each new gray node $y \in C(x)$ sets $P(y) = x$. At last, for all $n \in L$, where L is a set of the newly colored nodes, n sends a $DELETE$ message to $z \in N(n)$. z deletes n from $C(z)$ if applicable. The algorithm is repeated from Step 2.

4.2.2 D-CDS-UBG: A Distributed CDS Computation Algorithm in UBGs

This algorithm exploits D-MIS-UBG to have an MIS, and based on the MIS, it computes a CDS of a UBG. The parent-children relationship built while executing D-MIS-UBG will be ignored and a new one will be established. At the end of execution, the set of blue nodes will be a CDS. A message between two blue nodes goes through the shortest path between them over the set of blue nodes.

1. Run D-MIS-UBG (G).
2. r broadcasts a $IGNORE$ message to all other nodes. A node x sets $C(x)$ the set of gray nodes adjacent to x after receiving the message.
3. r colors itself blue, and every node $x \in C(r)$ sets $P(x) = r$ and $C(x) = C(x) \setminus N[r]$.
4. r broadcasts an ASK message to gray nodes $y \in N(x)$ for every blue node x. That is, r sends the message to x first and x broadcasts the messages to its gray neighbors. When y receives the message, it sends back an $REP(x, b(x))$ message, where $b(x)$ is the number of black nodes adjacent to x.
5. For every blue node x which is on the path between the sender of a reply message and r, x waits for reply messages from all nodes in $C(x)$. Then, x picks one node $y \in C(x)$ with maximum $b(y)$, sets $Index(x) = y$ and $b(x) = b(y)$, and sends an $REP(x, b(x))$ message to $P(x)$.
6. When the root r receives REP messages from all nodes in $C(r)$, it picks one child $x \in C(r)$ such that $b(x)$ is maximum. If $b(x) = 0$, then the set of blue nodes is a CDS and the computation is finished by broadcasting an FIN message through the CDS. If $m(x) > 0$, then r sends a $JOIN$ message to x.
7. On receiving a $JOIN$ message, if the node x is blue, then it forwards the $JOIN$ message to the node $Index(x)$; if the node x is gray, then x colors itself blue and all black nodes adjacent to x are also colored blue, $P(x)$ is set to be the node that sends $JOIN$ message to x, each other new blue node y sets $P(y) = \{x\}$. At last, for all $n \in L$, where L is a set of the new blue nodes, n sends a $DELETE$ message to $z \in N(n)$. z deletes n from $C(z)$ if applicable. The algorithm is repeated from Step 4.

Theorem 4.6. *Both the time and message complexities of D-CDS-UBG are $O(n^2)$, where n is the number of nodes in a given input UBG.*

Proof. The overall behavior of both D-MIS-UBG and the rest part of D-CDS-UBG is similar. First, both of them are executed for several rounds. In each round, one node in a subset sends a message to nodes adjacent to the subset, and later one preferred node among the neighbors of the subset is selected by the initiator and added to the subset. Therefore, both the time and message complexities of each round are equal to $O(n)$. Since we can have at most $O(n)$ rounds, the total time and message complexities of D-CDS-UBG are $O(n^2)$. □

5 Simulation Result

In this section, we compare D-CDS-UBG with the only and the best distributed algorithm by Butenko and Ursulenko to solve MCDS in UBGs [7]. We do not consider Zhong et al.'s work [14] for our comparison due to their problems as we mentioned earlier in Section 1. For more detail, see the Appendix. We do not consider Zou et al.'s work since their algorithm does not work in distributed environments [9]. For the simulations, we prepare a 100 × 100 × 100 three-dimensional virtual space and deploy wireless nodes. The number of nodes varies from 50 to 150 by increasing 10. We use 20, 25, 30, or 35 as the maximum transmission range of the nodes. Under the same parameter setting, 100 connected UBGs are randomly generated, a CDS of each graph is computed, and an average CDS size is calculated for each algorithm. In both Figs. 9 and 10, we use the following notations to represent each graph: We denote a solution

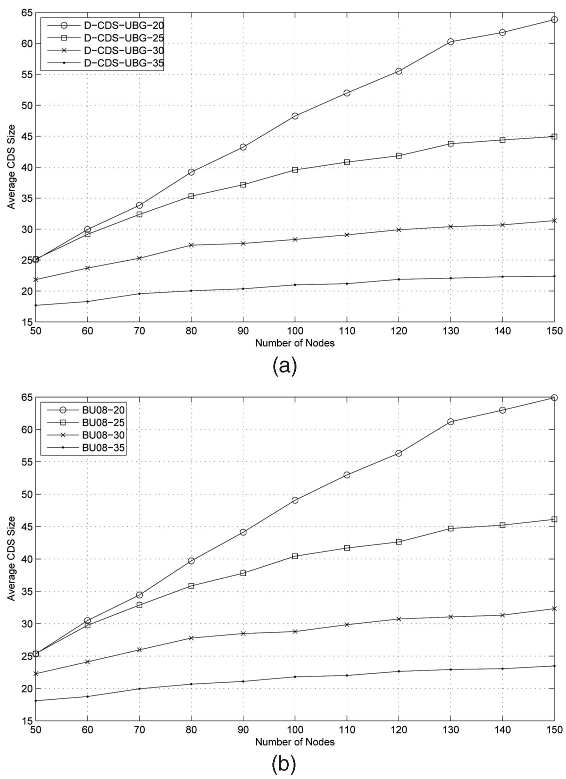


Fig. 9. Both algorithms generate smaller CDSs as the maximum transmission range increases. In addition, the size of the CDSs increases as the size of the input graphs grows.

obtained using our distributed algorithm by D-CDS-UBG. BU08 represents a graph computed using Butenko and Ursulenko's. The last number represents the maximum transmission range of each node used for generating that graph, which can be 20, 25, 30, or 35.

Fig. 9 shows how the performance of each algorithm changes as the maximum transmission range grows and as the size of network increases. Both algorithms generate the largest CDSs, on average, when the maximum transmission range is 20. As the transmission range increases, the average size of CDSs decreases. This is natural since with a larger maximum transmission range, one CDS node can dominate more non-CDS nodes, and therefore, we need less nodes to construct a CDS. In addition, as the size of the network grows, the average size of CDS should increase exactly like in the figure since we need a bigger CDS to dominate more non-CDS nodes.

Fig. 10 shows the performance comparison of both algorithms. Since the basic idea of D-CDS-UBG to compute an MIS is same with that of BU08's, the same MISs are used as inputs of both algorithms. Thus, the difference on the performance of both algorithms comes from their greedy strategies. When the maximum transmission range is 20, D-CDS-UBG is slightly better than BU08. The performance gap between them grows as we increase the maximum transmission ranges. We can also learn that the performance gap becomes apparent as the size of the network grows. From these results, we can conclude that 1) D-CDS-UBG works better than BU08 in general and 2) as the density of a network goes up, their performance gap becomes larger.

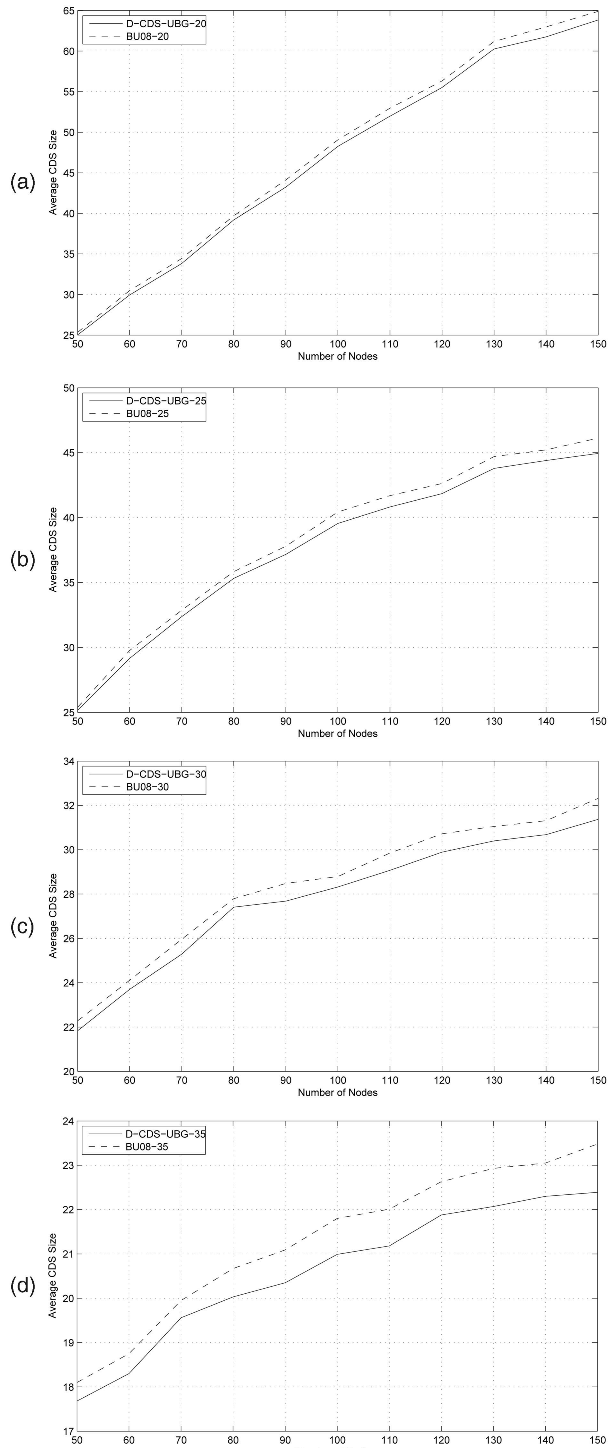


Fig. 10. Under any parameter setting, our algorithm works better than Butenko and Ursulenko's, on average. The average performance gap between them grows as the size and density of the networks increase.

6 Conclusion

In this paper, we study the problem of constructing minimum VBs in a distributed manner for 3D homogeneous wireless networks such as USNs. This problem can be abstracted as MCDS problem in UBGs, but it is NP-hard to get an optimal solution. To get an approximated solution, we improve the upper bound of size of an MIS in UBGs and

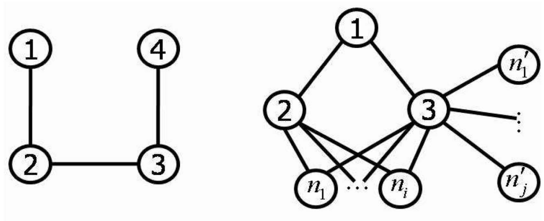


Fig. 11. Examples show the problem of Zhong et al.'s work.

present two new approximation algorithms, C-CDS-UBG and D-CDS-UBG. We introduce a centralized algorithm C-CDS-UBG first from which we build a distributed algorithm D-CDS-UBG. Both algorithms generate a CDS by computing an MIS first and connecting the MIS by adding some more nodes to the MIS. We prove that the PR of CDS-UBG (C-CDS-UBG and D-CDS-UBG) is 14.937, which is better than most recent work by Butenko and Ursulenko [7]. In simulation, we show that D-CDS-UBG outperforms the competitor's algorithm, which coincides with our theoretical analysis.

Now, we present some future work. First, while getting a tighter upper bound of an MIS of a UBG, we found that there is a gap between its current lower bound and upper bound. Therefore, we are interested in further reducing this gap. Second, even though the PR of D-CDS-UBG is better than the previous work, we think that the PR of D-CDS-UBG, 14.937, is still huge, and therefore, we believe that it can be reduced further.

Appendix

Problems with the Algorithm in [14]

The algorithm in [14], ZWH07 in short, is a two-phase distributed algorithm. It first computes an MIS and connects the nodes in it to generate a CDS. A node becomes an MIS node if all neighbors with higher weight become non-MIS nodes because they have an MIS neighbor. Note that, in this case, the hop distance between any two MIS nodes is at most three hops. Once an MIS is selected, ZWH07 connects the MIS nodes using the same greedy strategy to D-CDS-UBG.

Now, suppose that we have Fig. 11a as an input for ZWH07. Then, nodes 1 and 4 can be selected as dominators and nodes 2 and 3 as dominatees. Also, suppose that node 1 initiates the second phase to connect dominators. Since node 2 has no neighboring dominators, it will report 0 as the number of adjacent dominators to node 1. If node 1 does not allow node 2 to be a connector, this algorithm will be terminated and its result is not a CDS. If node 1 allows this, it incurs another problem as follows: Suppose that we have Fig. 11b as an input and nodes 2 and 3 are selected as dominators. Then they can be connected by node 1. However, since we allow any dominatee with no neighboring noncon-dominator to be a connector, $n_1, \ldots, n_i$ will be selected. Furthermore, terminal nodes $n'_1, \ldots, n'_j$ will be selected eventually, which implies that the resulting CDS can be arbitrarily larger than an optimal CDS, where a noncon-dominator is an MIS node that is not connected to the connected subset yet.

Problems with the Approximation Analysis in [14]

The approximation ratio of the algorithm by Zhong et al. depends on the following two claims [14]: First, in a UBG, a node has at most 11 neighbors, which are independent from each other. Second, when two nodes x and y are adjacent, at most eight independent neighbors of x are not adjacent to y.

The first claim is incorrect: a node can have 12 independent neighbors. In fact, suppose a regular icosahedron has a unit ball as its circumscribed sphere, then the edge length of the icosahedron is $4/\sqrt{10+2\sqrt{5}} \approx 1.051$, which implies that any two vertices of the icosahedron are at least 1.051 away from each other. Hence, if we put a node on the location of each vertex of the icosahedron, we can put 12 nodes which are independent from each other and adjacent to the center of the unit ball.

The second claim is also incorrect: there may exist two adjacent nodes x and y in a UBG such that x has 11 independent neighbors which are not adjacent to y. In fact, suppose that $d(x, y) = 1$, and denote the unit balls centered at x and y by B_x and B_y, respectively. Embed a regular icosahedron in B_x such that one vertex v of the icosahedron coincides with y. Because any point in $B_x \cap B_y$ has distance at most 1 from y, we see that the remaining 11 vertices of the icosahedron are outside of $B_x \cap B_y$ since they are at least 1.051 away from v. It follows that these 11 vertices are independent neighbors of x that are not adjacent to y.

If we apply the above corrected data to the proof in Zhong et al.'s work, the approximation ratio of their algorithm is in fact 22, which is equal to Butenko and Ursulenko's result.

Acknowledgments

This work was supported in part by the US National Science Foundation (NSF) under Grant nos. IIS-0513669, CCF-0621829, and CNS-0524429. This research was jointly supported by MEST, Korea, under WCU (R33-2008-000-10044-0); by a KRF Grant (KRF-2008-314-D00354); and by MKE, Korea, under ITRC IITA-2009-(C1090-0902-0046) and IITA-2009-(C1090-0902-0007). It was also supported in part by the NSFC (60603003), the Key Project of the Chinese Ministry of Education (208161), and the Scientific Research Program of the Higher Education Institution of Xinjiang. The authors would like to appreciate the time and effort of the editors and reviewers for their comments to improve the quality of this paper.

References

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

[2] A. Ephremides, J. Wieselthier, and D. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proc. IEEE*, vol. 75, no. 1, pp. 56-73, Jan. 1987.

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

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

[5] M. Cardei, X. Cheng, X. Cheng, and D.-Z. Du, "Connected Domination in Multihop Ad Hoc Wireless Networks," *Proc. Int'l Conf. Computer Science and Informatics,* Mar. 2002.

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

[7] S. Butenko and O. Ursulenko, "On Minimum Connected Dominating Set Problem in Unit-Ball Graphs," Nov. 2007, submitted.

[8] W. Wu, H. Du, X. Jia, Y. Li, and S. Huang, "Minimum Connected Dominating Sets and Maximal Independent Sets in Unit Disk Graphs," *Theoretical Computer Science,* vol. 352, pp. 1-7, 2006.

[9] Z. Zhang, X. Gao, W. Wu, and D. Du, "PTAS for Minimum Connected Dominating Set in Unit Ball Graph," *Proc. Third Int'l Conf. Wireless Algorithms, Systems and Applications (WASA '08),* Oct. 2008.

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

[11] M. Li, P.-J. Wan, and F. Yao, "Tighter Approximation Bounds for Minimum CDS in Wireless Ad Hoc Networks," *Proc. 20th Int'l Symp. Algorithms and Computation (ISAAC '09),* Dec. 2009.

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

[13] I.F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *J. Ad Hoc Network,* vol. 3, no. 3, pp. 257-279, Mar. 2005.

[14] X. Zhong, J. Wang, and N. Hu, "Connected Dominating Set in 3-Dimensional Space for Ad Hoc Network," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '07),* Mar. 2007.

[15] F. Zou, X. Li, D. Kim, and W. Wu, "Construction of Minimum Connected Dominating Set in 3-Dimensional Wireless Network," *Proc. Third Int'l Conf. Wireless Algorithms, Systems and Applications (WASA '08),* Oct. 2008.

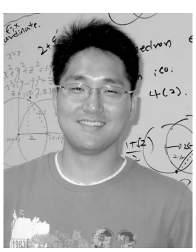
[16] C. Zong, *Sphere Packings.* Springer-Verlag, 1999.

[17] J.C. Hansen and E. Schmutz, "Comparison of Two CDS Algorithms on Random Unit Ball Ggraphs," *Proc. Seventh Workshop Algorithm Eng. and Experiments (ALENEX '05),* Jan. 2005.

[18] R. Hoppe, "Bemerkungen der Redaktion," *Grunert Arch. Math. Phys.,* vol. 56, pp. 307-312, 1874.

[19] H. Huang, A.W. Richa, and M. Segal, "Approximation Algorithms for the Mobile Piercing Set Problem with Applications to Clustering in Ad-Hoc Networks," *ACM Springer Mobile Networks and Applications (MONET),* vol. 9, no. 2, pp. 151-161, 2004.

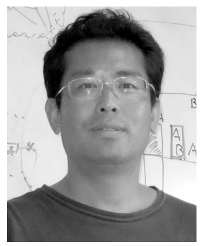
[20] K.H. Rosen, *Discrete Mathematics and Its Applications,* fifth ed. McGraw-Hill Higher Education, 2003.



Donghyun Kim received the BS degree in electronic and computer engineering from Hanyang University, Ansan, Republic of Korea, in 2003, the MS degree in computer science and engineering from Hanyang University, Republic of Korea, in 2005, and the PhD degree in computer science from the University of Texas at Dallas in 2010. His research interests include wireless networks, mobile computing, distributed computing, approximation algorithm design and analysis, combinatorial optimization, and network security. He is a student member of the IEEE.

Zhao Zhang received the PhD degree from Xinjiang University in 2003. She is a full professor in the College of Mathematics and System Sciences at Xinjiang University. Her current research interests include combinatorial optimization and graph theory.

Xianyue Li received the BS and PhD degrees from Lanzhou University, China, in 2003 and 2009, respectively. He is currently an assistant professor in the School of Mathematics and Statistics at Lanzhou University, China. His research interests include wireless networks, graph theory, and approximation algorithm design and analysis.



Wei Wang received the BS degree from ZheJiang University, China, in 1991, and the MS and PhD degrees from Xi'an Jiaotong University, China, in 1994 and 2006, respectively. He is currently an assistant professor in the Department of Mathematics at Xi'an Jiaotong University. His research interests include algebraic graph theory and approximation algorithm design and analysis.

Weili Wu received the PhD degree in computer science from the University of Minnesota in 2002. She is currently an associate professor in the Department of Computer Science at the University of Texas at Dallas. Her main research interest is in database systems. She has produced a number of research papers in spatial data mining, distributed database systems, and algorithm design. She was also a coeditor of a book on clustering and information retrieval. She is a 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 in 1985, under the supervision of Professor Ronald V. Book. Before settling at the University of Texas at Dallas, he worked as a professor in the Department of Computer Science and Engineering, University of Minnesota. He also worked at the Mathematical Sciences Research Institute, Berkeley, for one year, in the Department of Mathematics, Massachusetts Institute of Technology, for one year, and in the Department of Computer Science, Princeton University, for one and a half years. He is the editor-in-chief of the *Journal of Combinatorial Optimization* and is also on the editorial boards for several other journals. Forty PhD students have graduated under his supervision. He is a member of the IEEE.