Optim Lett (2010) 4:321–333
DOI 10.1007/s11590-009-0148-3

ORIGINAL PAPER

A PTAS for minimum d-hop connected dominating set in growth-bounded graphs

Xiaofeng Gao · Wei Wang · Zhao Zhang · Shiwei Zhu · Weili Wu

Received: 15 June 2009 / Accepted: 23 September 2009 / Published online: 9 October 2009

Abstract In this paper, we design the first polynomial time approximation scheme for d-hop connected dominating set (d-CDS) problem in growth-bounded graphs, which is a general type of graphs including unit disk graph, unit ball graph, etc. Such graphs can represent majority types of existing wireless networks. Our algorithm does not need geometric representation (e.g., specifying the positions of each node in the plane) beforehand. The main strategy is clustering partition. We select the d-CDS for each subset separately, union them together, and then connect the induced graph of this set. We also provide detailed performance and complexity analysis.

Keywords PTAS · Connected dominating set · Growth-bounded graph

X. Gao (✉) · W. Wu
Department of Computer Science, University of Texas at Dallas, Richardson, USA
e-mail: xxg052000@utdallas.edu

W. Wu
e-mail: weiliwu@utdallas.edu

W. Wang
School of Science, Xi'an Jiaotong University, Xi'an, People's Republic of China
e-mail: wang_weiw@163.com

Z. Zhang
College of Mathematics and System Sciences, Xinjiang University, Xinjiang, People's Republic of China
e-mail: zhzhao@xju.edu.cn

S. Zhu
Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA
e-mail: josh.zhu@gmail.com

1 Introduction

Wireless networks, such as wireless ad hoc networks, wireless sensor networks, and wireless telecommunications networks, have numerous applications in the world including both military applications (e.g., battlefield surveillance, enemy detection and environmental monitoring) and civilian applications (e.g., cellular phones, emergency services and healthcare management). They use electromagnetic waves to exchange messages, and form wireless information transmission system. The development of wireless networks attracts more and more attention from both scientists and engineers for their potential implementations.

However, due to limited radio range and power constraint, most of the nodes in wireless network cannot communicate with distant nodes directly, but exchange data via a routing path through several intermediate nodes. Moreover, without a well-organized routing scheme, wireless networks always use simple flooding to transmit messages, bringing high energy consuming and interference problems. To improve the performance and Quality of Service requirement, a special subset of nodes in the network is selected to form a *virtual backbone*, which will participate in multi-hop routing process. Usually, a Connected Dominating Set (CDS) is the graph model of such subsets. Each node in CDS (dominator) will dominate its adjacent neighbors (dominatees). Thus the network is divided as several clusters, each with its dominator as the clusterhead.

Later, researchers found that for large-scaled wireless network, only fixing a CDS is not enough to build a hierarchical based, energy efficient communication system. Thus they generalized the concept of CDS into a more powerful expression, the d-Hop CDS (d-CDS). The formal definition of d-CDS is: given a graph $G = (V, E)$, a d-hop connected dominating set (d-CDS) is a vertex subset D of V such that for any vertex u, either $u \in D$ or there exists a vertex $v \in D$, and we can find a path in G from u to v within d hops (or this path has $d - 1$ intermediate nodes). Besides, the subgraph induced by $G[D]$ is connected. If only consider dominating properties, the selected subset is called d-hop dominating set (d-DS) for short.

Figure 1 is an example of 2-CDS in undirected graph, where grey regions form a group of clusters, red nodes in each cluster are the clusterheads (also form 2-DS), and

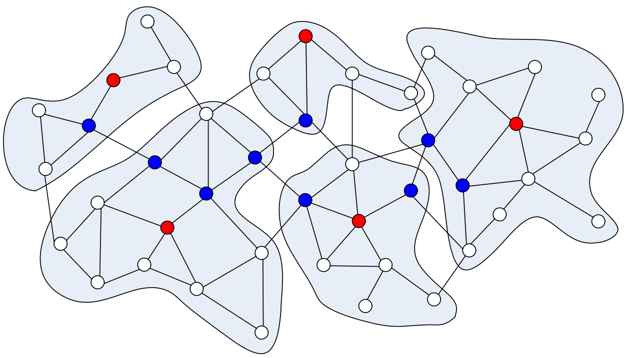

Fig. 1 An example of a 2-CDS in an undirected graph

blue nodes make them connected. The union of red and blue nodes form a 2-CDS together.

It is easy to know that the CDS we talked previously is 1-CDS according to this definition. d-CDS is more powerful to partition the network into bigger cluster, and each node is the super-clusterhead, which can be viewed as the base station to transfer huge amount of information. Therefore, each cluster can be shrunk as a supernode in a higher level network. Such multi-level hierarchical based wireless network architecture highly simulates wired network transmissions among wide ranges, bringing better performance reliability and system capabilities. As large scaled wireless networks are used in more and more fields, finding a d-CDS attracts more attention from academia and industry.

1.1 Previous literatures

Most of the related works on d-CDS are completed within recent ten years, especially after year 2000. Vuong et al. [13] proved that finding minimum d-CDS is *NP*-complete in general graph by a reduction from 3-SAT, and later Nguyen [7] proved that d-CDS remains *NP*-complete in UDG. Researchers are looking for effective approximation algorithms to find a feasible solution within polynomial time. Many previous literatures [1,3,4,9–12,14] provided heuristic algorithms to solve general d-CDS problem or relative problems with fixed d. They also provided simulations to show their performance. However, most of them lacked approximation analysis, especially the quality of their approximate solutions. More recently, Gao et al. [5] provided the first constant-factor approximation algorithm to solve d-CDS in unit disk graph, which is a distributed algorithm practical for many applications. Their algorithm has approximation ratio around $0.335(d + 0.5)^3$ with some fixed d.

Cheng et al. [2] proposed the first Polynomial Time Approximation Scheme (PTAS) for minimum CDS problem in unit disk graph. A PTAS is a $(1 + \varepsilon)$-approximation, where ε is an arbitrary positive number. It means that the result of this algorithm can be really close to the optimal solution, as what ε you required before algorithm runs. Their main strategy is grid partition on 2-dimensional space and shifting policy to reduce the size of additional nodes to connect the whole graph. Later, Nieberg and Hurink [8] found another partition method to design a PTAS for minimum Dominating Set (DS) problem in unit disk graph. They defined *clustering partition* to divide the whole graph into several clusters according to some prefixed parameters, and later Gfeller and Vicari [6] modified their method to deal with minimum CDS problem in growth-bounded graphs. Till now, no literature provides PTAS for d-CDS problem yet.

1.2 Our contribution

In this paper, we present the first PTAS for minimum d-CDS in growth-bounded graphs (including unit disk graph, unit ball graph, etc), which can represent the majority types of wireless networks in practice. The major strategy we used is clustering partition. Firstly, we divide the whole graph into clusters, and then for each cluster we select a d-CDS as subsolution. Finally we union these subsolutions and add some nodes to make the final result connected. Based on the specific characteristics of d-CDS, we fix several intermediate parameters to bound the result of our algorithm within a controllable range, and provide performance analysis to show the correctness and robustness.

The rest of our paper is organized as follows. Section 2 introduces some useful conceptions and theorems. Section 3 describes the algorithm design and formula for the intermediate parameters. In Sect. 4 we illustrate the correctness and performance of our algorithm, including time complexity and approximation ratio analysis. Finally, in Sect. 5 we give a conclusion and discuss about future works in this field.

2 Preliminaries

In this section we introduce some concepts which are helpful for later discussions. Given an undirect graph $G = (V, E)$ as a network, $\forall v \in V$, let $N(v)$ denote the set of adjacent neighbors of v, and $N^r(v)$ the set of r-hop neighbors of v (nodes that reach v via at most $r-1$ intermediate nodes). Obviously, $N(v) = N^1(v)$. For $S \subseteq V$, let $N^r(S) = \cup_{v \in S}\{N^r(v)\}$. Let $C_d(S)$ denote the d-CDS for $S \subseteq V$, and then $C_d(S) \subseteq N^d(S)$.

When selecting a CDS, some literatures use two-step greedy algorithms: (1) find a Maximal Independent Set (MIS); (2) connect this selected MIS. An MIS is automatically a Dominating Set (DS) and it is much easier to select. Similarly, when considering d-CDS problem, we can find a d-hop MIS (d-MIS) first and then try to connect this subset. The formal definition of d-MIS can be listed as follows.

Definition 1 (d-MIS) A d-MIS for $G = (V, E)$ is a d-hop independent set $M \subseteq V$ such that $\forall u, v \in M$, there does not exist a path between them within d hops (say, a path P with at most $d-1$ intermediate nodes, $p = (u, i_1, i_2, \ldots, i_{d-1}, v)$. Moreover, if we insert any node from $V \backslash M$ into M, then M is not an independent set any more.

Easy to see, 1-MIS is MIS for short. We use $MIS_d(S)$ to represent a d-MIS for a vertex subset. Then, $MIS(S) = MIS_1(S)$. It is trivial that $|MIS_d(S)| \leq |MIS(S)|$. Thus any upper bound obtained for $|MIS(S)|$ is an upper bound for $|MIS_d(S)|$. For any node sets $S \subseteq R$, $|MIS(S)| \leq |MIS(R)|$. These two observations will be used frequently in the paper without mentioning it explicitly.

Definition 2 (*Growth-bounded graph*) A graph G is *growth-bounded* if there exists a polynomial function $f(r)$ such that $|MIS(N^r(v))| \leq f(r)$ for each $v \in V$, where f only depends on r and is independent of the structure of graph G.

Unit Disk Graph (UDG), Unit Ball Graph (UBG) etc. all belong to growth-bounded graphs. For example, Fig. 2 shows that UDG has $f(r) \leq (2r+1)^2$.

3 Algorithms

Before introducing our algorithm, we need to briefly summarize the works done by Nieberg et al. [8] and Gfeller et al. [6] since their works are our fundamentals.

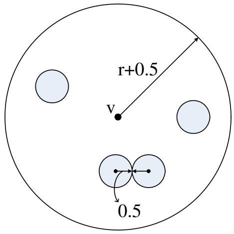

Fig. 2 For a node v in UDG with radius 1, if we shrink the radius of other nodes into 0.5, then all the neighbors of v in $N^r(v)$ should locate within a circle of v with radius $r + 0.5$. Moreover, every two nodes u, w in $MIS(N^r(v))$ should not intersect each other since $dist(u, w) \geq 1$. Thus, the maximum $MIS(N^r(v))$ should be less than the area of big circle divided by the area of small disks. Say, $|MIS(N^r(v))| \leq \frac{\pi\cdot(r+0.5)^2}{\pi\cdot 0.5^2} = (2r + 1)^2 = f(r)$

3.1 Discussions on previous works

In [8], Nieberg et al. proposed a PTAS to compute a MDS in a UDG. The main idea is: a collection of subsets $\{S_1, \ldots, S_k\}$ is defined to be *a 2-separated partition of G* if the distance $dist(S_i, S_j) > 2$ holds for any $i \neq j(dist(S_i, S_j) = \min_{u\in S_i, v\in S_j}\{dist(u, v)\}$. This partition denotes that $N(S_i)\cap N(S_j) = \emptyset$, such that we can select DS for each S_i without repeatedly choosing one node). $D(S)$ is a minimum dominating set of S. Note that $D(S) \subseteq N(S)$. For a 2-separated partition $\{S_1, \ldots, S_k\}$ of G, $D(S_1), D(S_2), \ldots, D(S_k)$ are disjoint. Combining this with the observation that $D(V) \cap N(S_i)$ is a dominating set of S_i, we have $\sum_{i=1}^{k} |D(S_i)| \leq |D(V)|$. Note that $\bigcup_{i=1}^{k} D(S_i)$ is not necessarily a dominating set of G. Hence they further enlarge the S_i's to T_i's such that (1) $S_i \subseteq T_i$; (2) $|T_i| \leq (1+\varepsilon)|S_i|$; and (3) $\bigcup_{i=1}^{k} T_i$ is a dominating set of $V(G)$.

Then $\bigcup_{i=1}^{k} T_i$ is a $(1 + \varepsilon)$-approximation of $D(V)$. To compute $\{S_1, \ldots, S_k\}$ and $\{T_1, \ldots, T_k\}$, they pick an arbitrary node v_1, find the minimum integer r_1 satisfying $|D(N^{r_1+2}(v_1))| \leq (1 + \varepsilon)|D(N^r(v_1))|$. Then set $S_1 = N^{r_1}(v_1)$ and $T_1 = N^{r_1+2}(v_1)$, repeat this procedure in the remaining graph $G[V\backslash T_1]$. For simplicity of statement, we name the nodes $v_1, v_2, \ldots$ picked in their algorithm as *core nodes*.

The above idea was further used by Gfeller et al. to give a PTAS for 1-CDS in grwoth-bounded graphs [6]. In their algorithm, T_i's are enlarged a little larger compared with Nieberg's algorithm so that the overlap of T_i's are large enough to ensure the connectivity of the output. Actually, each T_i can be viewed as a cluster, and this kind of partition is named as *clustering partition*. Compared with grid partition, the kernel of clustering partition is to fix parameter ε such that the redundancies (extra selected nodes compared with the optimal solution) can be controlled within a very small range.

3.2 Algorithm description

In this paper, we gave a PTAS computing a d-CDS in a growth-bounded graph. The main line of the proof follows that of [6,8]. However, the algorithm in [6] can not be directly generalized for d-CDS by simply replacing '1' by 'd', since the output might not be connected. Hence to solve d-CDS, some new ideas are needed. Firstly, we have to add more nodes to make the output connected. Secondly, the number of nodes added must be small enough to guarantee the $(1+\varepsilon)$-approximation. For this purpose, the positions of the core nodes chosen from the algorithm do matter (in [6,8], the core nodes are chosen arbitrarily). The details can be seen from Algorithm 1.

Algorithm 1 PTAS for d-CDS

Input: An abstract growth-bounded graph G, parameters d and ε.
Output: A d-hop CDS C.

1: Set $U = V(G)$, $k = 0$.
2: Choose a node $u \in U$; find the smallest integer r satisfying $|MIS_d(N^{r+d}(u)\backslash N^r(u))| \leq \varepsilon|MIS_d(N^r(u))|$. Let $W = N^{r+3d}(u)\backslash N^{r+2d}(u)$. Push (u, W) into the queue Q. Set $k = k+1$, $v_k = u$, $S_k = N^r(u)$, $T_k = N^{r+3d}(u)$. Set $U = U\backslash N^{r+2d}(u)$.
3: **while** $U \neq \emptyset$ **do**
4: Pop a pair $(\tilde{u}, \tilde{W})$ from Q. Set $Child(\tilde{u}) = \emptyset$.
5: **while** $\tilde{W} \neq \emptyset$ **do**
6: Choose a node $u \in \tilde{W}$; find the smallest r s.t. $|MIS_d(N^{r+d}(u)\backslash N^r(u))| \leq \varepsilon|MIS_d(N^r(u))|$. Let $W = N^{r+3d}(u)\backslash N^{r+2d}(u)$. Push (u, W) into the end of the queue Q. Set $k = k+1$, $v_k = u$, $S_k = N^r(u)$, $T_k = N^{r+3d}(u)$, $U = U\backslash N^{r+2d}(u)$, $Child(\tilde{u}) = Child(\tilde{u}) \cup \{u\}$, $\tilde{W} = \tilde{W}\backslash N^{r+2d}(u)$.
7: **end while**
8: **end while**
9: For each $i = 1, \ldots, k$, compute a d-CDS C_i of T_i.
10: Set $C = \bigcup_{i=1}^{k} C_i$
11: **for** $i = 1, \ldots, k$ **do**
12: For each child $v_j \in Child(v_i)$, if C_i and C_j are not connected in $G[C]$, then find a shortest path in G connecting them and add the internal nodes of the path into C.
13: **end for**
14: Output C.

In Algorithm 1, a queue Q is used to record the core nodes, and k is used to record the number of core nodes which have been chosen. The symbols $N^r(u)$ etc. in the algorithm are referring to neighbor sets in $G[U]$ (not in G), where U is used to record nodes waiting to be dealt with. We call the node set $N^{r+3d}(u)\backslash N^{r+2d}(u)$ in line 2 and line 6 the *boundary area of* u, which is recorded by W. Note that except for the first core node v_1 chosen in line 2, all the other core nodes chosen in line 6 are from the boundary area of the popped node $\tilde{u}$. This is to ensure that by adding *relatively small number* of nodes, the output C can be connected. The core nodes chosen from the boundary area of v are called *children of* v. Actually, the core nodes form a tree in graph G, and the paths inserted into C are just used for connectivity with some intermediate nodes.

In the next section, we prove that the final output of Algorithm 1 returns a correct feasible solution with approximation ratio $(1+\varepsilon)\cdot OPT$, where OPT is the number of nodes in optimal solution. Besides, we also guarantee that the whole process can terminate within polynomial time.

4 Performance

4.1 Correctness and time complexity

Theorem 1 *The output C of Algorithm 1 is a d-CDS for given graph G.*

Proof In line 9, a d-CDS C_i is computed for each T_i. We do this by exhaust search (Lemma 1 guarantees that this can be done in polynomial time). Since $\bigcup_{i=1}^{k} T_i = V(G)$ (each T_i may overlap with some T_j and every node in U should belong to some T_i), we see that $C = \bigcup_{i=1}^{k} C_i$ is a d-hop DS for G. However, C might be disconnected. Hence in the second part (line 11 to 13), we connect C_j's to C_i whenever v_j is a child of v_i and C_i, C_j are not connected already. Clearly, the output C is a feasible d-CDS of G.

The following lemma plays an important role in our proofs. The intuition is that when r is sufficiently large, then $N^{r+d}(v)\backslash N^r(v)$ is so thin compared with $N^r(v)$ such that it contains relatively small number of independent nodes.

Lemma 1 *Let G be a growth-bounded graph with growth function f. For any positive integer d and any real number $\varepsilon > 0$, there exists a constant $r(d, \varepsilon, f)$ which depends on d, ε and f, but is independent of the topology of G, such that $|MIS_d(N^{r+d}(v)\backslash N^r(v))| \leq \varepsilon |MIS_d(N^r(v))|$ for each $v \in V(G)$.*

Proof We first show that for any integer $r \geq d$,

$$|MIS_d(N^{r+d}(v)\backslash N^r(v))| \leq f(2d)|MIS_d(N^r(v)\backslash N^{r-d}(v))|. \tag{1}$$

In fact, for each node $u \in N^{r+d}(v)\backslash N^r(v)$, there exists a node $w \in N^r(v)\backslash N^{r-d}(v)$ which is at most d hops away from u. Since w is d-hop dominated by some node in $MIS_d(N^r(v)\backslash N^{r-d}(v))$, we see that u is no more than 2d-hops away from a node in $MIS_d(N^r(v)\backslash N^{r-d}(v))$. Then inequality (1) follows from Lemma 2.

Suppose the lemma does not hold. Then there exists a node v such that $|MIS_d(N^{r+d}(v)\backslash N^r(v))| > \varepsilon |MIS_d(N^r(v))|$ for all $r \geq 0$, and thus

$$\begin{aligned} |MIS_d(N^r(v)\backslash N^{r-d}(v))| &\geq \frac{1}{f(2d)}|MIS_d(N^{r+d}(v)\backslash N^r(v))| \\ &> \frac{\varepsilon}{f(2d)}|MIS_d(N^r(v))|. \end{aligned}$$

Combining this with $|MIS_d(N^r(v))| \geq |MIS_d(N^r(v)\backslash N^{r-d}(v))| + |MIS_d(N^{r-2d}(v))|$ and $|MIS_d(N^r(v))| \geq |MIS_d(N^{r-2d}(v))|$, we have

$$\begin{aligned} |MIS_d(N^r(v))| &\geq |MIS_d(N^r(v)\backslash N^{r-d}(v))| + |MIS_d(N^{r-2d}(v))| \\ &> \frac{\varepsilon}{f(2d)}|MIS_d(N^r(v))| + |MIS_d(N^{r-2d}(v))| \end{aligned}$$

$$> \frac{\varepsilon}{f(2d)}|MIS_d(N^{r-2d}(v))| + |MIS_d(N^{r-2d}(v))|$$
$$> (1+\bar{\varepsilon})|MIS_d(N^{r-2d}(v))|, \tag{2}$$

where $\bar{\varepsilon} = \varepsilon/f(2d)$. By taking $r = 2id$ and recursively using inequality (2), we have

$$|MIS_d(N^{2id}(v))| \geq (1+\bar{\varepsilon})^i |MIS_d(N^0(v))| = (1+\bar{\varepsilon})^i. \tag{3}$$

On the other hand, since G is growth-bounded, we have

$$|MIS_d(N^{2id}(v))| \leq |MIS(N^{2id}(v))| \leq f(2id). \tag{4}$$

Since f is a polynomial function on i, we see that (3) and (4) contradict with each other when i is sufficiently large. Hence the lemma is true.

Lemma 1 shows that the minimum integer r in Algorithm 1 exists and is upper bounded by a constant $r(d, \varepsilon, f)$. Denote by r_i the r found in the ith iteration. Then $S_i = N^{r_i}(v_i)$. By the definition of growth-bounded graph, we can have the following relation.

Lemma 2 *Let S and R be two node sets of a growth-bounded graph G with growth function f such that every node in S is at most r-hops away from a node in R. Then* $|MIS(S)| \leq f(r)|R|$.

Proof Since $\forall v \in R$, the maximum number of MIS in $N^r(v)$ is bounded by $f(r)$, and there are totally $|R|$ nodes, combining $S \subseteq N^r(R)$, we have $|MIS(S)| \leq f(r)|R|$.

Corollary 2 *For any integers d_1, d_2 with $0 \leq d_1 < d_2 \leq 3d$, $|MIS(N^{d_2}(S_i)\backslash N^{d_1}(S_i))| \leq \varepsilon_1 |MIS_d(S_i)|$, where $\varepsilon_1 = f(3d)\varepsilon$ only depends on d, ε and f.*

Proof Since any node in $N^{d_2}(S_i)\backslash N^{d_1}(S_i)$ is at most $d_2 - d$ hops away from $N^d(S)\backslash S$, it is at most d_2 hops away from $MIS_d(N^d(S)\backslash S)$. Then by Lemmas 1 and 2, we have

$$|MIS(N^{d_2}(S_i)\backslash N^{d_1}(S_i)| \leq f(d_2)|MIS_d(N^d(S_i)\backslash S_i)| \leq f(3d)\varepsilon|MIS_d(S_i)|.$$

It is well known that for a DS D of a connected graph G, if $G[D]$ is not connected, then there exist two connected components of $G[D]$ which are at most 3-hops away from each other. This result plays an important role in extending a DS into a CDS while keeping performance ratio in control. For the d-hop case, a similar result can be obtained easily by a similar line.

Lemma 3 *Let G be a connected graph and D be a d-hop DS of G. If $G[D]$ is not connected, then there exist two connected components of $G[D]$ which are at most* $2d + 1$ *hops away from each other.*

However, in the following, we have to deal with such a case that a d-hop DS D *of a node set* S is not contained in S. In this case, similar result still holds, but the proof is a little different.

Lemma 4 *Suppose S is a node set of G such that $G[S]$ is connected, $D \subseteq N^d(S)$ is a d-hop DS of S. If $G[D]$ is not connected, then there exist two connected components of $G[D]$ which are at most $2d+1$ hops away from each other.*

Proof We may assume that every connected component of $G[D]$ d-hop dominates some nodes in S. Otherwise, we can delete those components which do not d-hop dominate any node in S, and find the desired components in the remaining ones.

For two connected components G_1, G_2 of G, we define the S-distance between G_1 and G_2, denoted by $dist_S(G_1, G_2)$, to be the minimum length of a G_1-to-G_2 path P such that

1. the first node u_1 on P which is in S is at distance not more than d from G_1;
2. the last node u_t on P which is in S is at distance not more than d from G_2;
3. the nodes on P between u_1 and u_t are all in S.

By our assumption in the above paragraph and the condition that $G[S]$ is connected, we see that $dist_S(G_1, G_2) < \infty$ for every pair of components. Let G_1, G_2 be two connected components of $G[D]$ with minimum $dist_S(G_1, G_2)$. Suppose $P = u_0u_1 \dots u_tu_{t+1}$ is a shortest G_1 to G_2 path whose internal nodes are all in S. If $t > 2d+1$, consider u_{d+1}. Since $u_{d+1} \in S$, it is d-hop dominated by a node $v \in D$. By the minimality of P, we have $v \in G_3$ where G_3 is a component of $G[D]$ different from G_1 and G_2. But then $dist_S(G_1, G_3) < dist_S(G_1, G_2)$, contradicting the choice of G_1 and G_2. Hence$dist_S(G_1, G_2) = t \leq 2d+1$. Fig. 3 is an counter example to illustrate this proof.

As a corollary of Lemma 4, extending a d-hop DS of a connected node set S can be done while the performance ratio is kept under control.

Corollary 3 *Suppose S is a node set of G such that $G[S]$ is connected, $D \subseteq N^d(S)$ is a d-hop DS of S such that $G[D]$ has s connected components and each connected component of $G[D]$ dominates some node in S. Then D can be extended to a d-hop CDS of S by adding at most $2d(s-1)$ nodes.*

By Lemma 1 and Corollary 3, we can prove that Algorithm 1 can be executed in polynomial time.

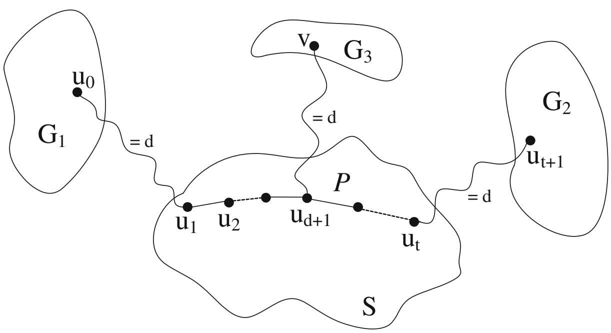

Fig. 3 An counter example to calculate the length of path P go through set S and connect two components G_1 and G_2 together

Theorem 4 *The time complexity of Algorithm 1 is $n^{O(r(d,\varepsilon,f)^2)}$, where n is the number of nodes in the graph G.*

Proof The most time consuming parts are line 2, 6, and 9, where exhaust searches are used.

It is easy to see that $|MIS_d(N^r(u))| \leq f(r)$ and $|MIS_d(N^{r+d}(u)\backslash N^r(u))| \leq f(r+d)$. Combining this with $r \leq r(d,\varepsilon,f)$, the exhaust searches in line 2 and 6 can be done in time $n^{O(r(d,\varepsilon,f)^2)}$.

Similarly, since $T_i = N^{r_i+3d}(v_i)$, we see that $D_i \triangleq MIS_d(T_i)$ is a d-hop DS of T_i with at most $[2(r_i+3d)+1]^2$ nodes. By Corollary 3, to connect them into a d-CDS of T_i requires adding at most $2d(|D_i|-1)$ nodes. Hence a d-CDS of T_i contains at most $(2d+1)[2(r(d,\varepsilon,f)+3d)+1]^2-2d$ nodes, and the exhaust search for it takes time $n^{O(r(d,\varepsilon,f)^2)}$. Thus our approximation can terminate within polynomial time.

4.2 Approximation ratio analysis

Next, we show that Algorithm 1 is a PTAS. First, the concept of 2-separated partition is generalized to the following:

Definition 3 ($2d$*-separated partition*) Let $S_1,\ldots,S_k$ be a collection of subsets of $V(G)$. If $dist(S_i,S_j) > 2d, \forall i \neq j$, then $S_1,\ldots,S_k$ is a $2d$-separated partition of G.

Clearly, the sets $\{S_1,\ldots,S_k\}$ in Algorithm 1 form a $2d$-separated partition of G. In the following, we use $C_d(S)$ to denote a d-CDS of node set S in G.

Lemma 5 *Let $\{S_1,S_2,\ldots,S_k\}$ be the 2d-separated partition of G obtained from Algorithm 1. Then $\sum_{i=1}^k |C_d(S_i)| \leq (1+\varepsilon_2)|C_d(V)|$, where $1+\varepsilon_2 = 1/[1-2df(d)\varepsilon_1]$.*

Proof Since $C_d(S_i) \subseteq N^d(S_i)$, we see that $C_d(V)\bigcap N^d(S_i)$ is a d-hop dominating set for S_i. Note that $G[C_d(V)\bigcap N^d(S_i)]$ might be disconnected. Suppose, without loss of generality, that every connected component of $G[C_d(V)\bigcap N^d(S_i)]$ d-hop dominates some node in S_i. By Corollary 3, $C_d(V)\bigcap N^d(S_i)$ can be extended to a d-hop CDS of S_i by adding at most $2d(s-1)$ vertices of G, where s is the number of connected components of $G[C_d(V)\bigcap N^d(S_i)]$. Hence

$$|C_d(S_i)| \leq |C_d(V)\cap N^d(S_i)| + 2d(s-1). \tag{5}$$

Since $G[C_d(V)]$ is connected, we see that for each connected component of $G[C_d(V)\cap N^d(S_i)]$, there exists a vertex $u \in C_d(V)\cap(N^d(S_i)\backslash N^{d-1}(S_i))$. These u's are clearly independent since they belong to different components. Hence by Corollary 2 and Lemma 2,

$$s \leq |MIS(N^d(S_i)\backslash N^{d-1}(S_i))| \leq \varepsilon_1|MIS_d(S_i)| \leq f(d)\varepsilon_1|C_d(S_i)|. \tag{6}$$

Combining inequalities (5) and (6) together, we have

$$|C_d(S_i)| < |C_d(V)\cap N^d(S_i)| + 2df(d)\varepsilon_1|C_d(S_i)|.$$

Adding them over $i = 1, \ldots, k$, by noting that $C_d(V) \cap N^d(S_i)$'s are disjoint, we have

$$\sum_{i=1}^{k} |C_d(S_i)| < |C_d(V)| + 2df(d)\varepsilon_1 \sum_{i=1}^{k} |C_d(S_i)|,$$

and the lemma follows.

Lemma 6 $|C_d(T_i)| \leq (1 + \varepsilon_3)|C_d(S_i)|$, *where* $\varepsilon_3 = (2d + 1) f(3d) f(d)\varepsilon$.

Proof By adding an $MIS_d(T_i \backslash S_i)$ to $C_d(S_i)$, we obtain a DS of T_i. Then by Lemma 3, adding at most $2d|MIS_d(T_i \backslash S_i)|$ nodes makes it connected. Hence

$$|C_d(T_i)| \leq |C_d(S_i)| + (2d + 1)|MIS_d(T_i \backslash S_i)|. \tag{7}$$

Since each node in $T_i \backslash S_i$ is at most $2d$-hops away from a node in $N^d(S_i)\backslash S_i$, and each node in $N^d(S_i)\backslash S_i$ is d-hop dominated by a node in $MIS_d(N^d(S_i)\backslash S_i)$, we see that each node in $T_i \backslash S_i$ is at most $3d$-hops away from a node in $MIS_d(N^d(S_i)\backslash S_i)$, and thus by Lemmas 1 and 2

$$\begin{aligned} |MIS_d(T_i \backslash S_i)| &\leq |MIS(T_i \backslash S_i)| \leq f(3d)|MIS_d(N^d(S_i)\backslash S_i)| \\ &\leq f(3d)\varepsilon|MIS_d(S_i)| \leq f(3d) f(d)\varepsilon|C_d(S_i)|. \end{aligned} \tag{8}$$

Then the lemma follows from combining inequalities (7) and (8).

Lemma 7 *The number of nodes added in line 12 of Algorithm 1 is no more than* $\varepsilon_4|C_d(V)|$ *where* $\varepsilon_4 = (2d - 1) f(d)\varepsilon_1(1 + \varepsilon_2)$.

Proof By the choice of the core nodes, all children of v_i are in $N^{3d}(S_i)\backslash N^{2d}(S_i)$ and are d-hop independent. Hence by Corollary 2 and Lemma 2,

$$|Child(v_i)| \leq |MIS_d(N^{3d}(S_i)\backslash N^{2d}(S_i))| \leq \varepsilon_1|MIS_d(S_i)| \leq f(d)\varepsilon_1|C_d(S_i)|.$$

Consider a child v_j of v_i. Since $v_j \in T_i \cap T_j$, there exist two nodes $u_i \in D_d(T_i) = C_i$ and $u_j \in D_d(T_j) = C_j$ both d-hop dominating v_j. Hence the distance between $G[C_i]$ and $G[C_j]$ is at most $2d$ hops. Since we use a *shortest* path to connect C_j to C_i in line 12, the number of nodes added is at most $2d - 1$. The total number of nodes added from line 11 to line 13 is upper bounded by

$$(2d - 1) \sum_{i=1}^{k} |Child(v_i)| \leq (2d - 1) f(d)\varepsilon_1 \sum_{i=1}^{k} |C_d(S_i)|.$$

Then the result follows from Lemma 5.

Now, we are ready to prove the performance ratio of Algorithm 1.

Theorem 5 *Algorithm 1 is a PTAS for the d-hop CDS in growth-bounded graphs.*

Proof The output C of Algorithm 1 is clearly a d-hop CDS of the input graph G. By Lemmas 5, 6 and 7, we have

$$|C| \leq \sum_{i=1}^{k} |C_d(T_i)| + \varepsilon_4 |C_d(V)| \leq [(1+\varepsilon_3)(1+\varepsilon_2)+\varepsilon_4]|C_d(V).$$

Taking $\bar{\varepsilon} = \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \varepsilon_2\varepsilon_3$, we have $|C| \leq (1+\bar{\varepsilon})|C_d(V)|$. Noting that $\bar{\varepsilon}$ is only dependent on d, ε and f, but is independent of the topology of graph G. Then our proof is completed.

5 Conclusions

In this paper, we proposed the first polynomial time approximation scheme (PTAS) to solve the d-hop connected dominating set problem (d-CDS) in growth-bounded graphs (including unit disk graph, unit ball graph, etc.). The main strategy we used is to partition the given graph into several clusters based on some preknown parameters, select d-CDS's for each cluster separately by exhaust search, and then union them together. If they are not connected, insert some additional nodes from shortest path between two connected components with parent and children relationships. We also tried to bound the size of the added nodes as small as possible compared with the optimal solution. Besides, we provide detailed analysis for algorithm performance including time complexity and approximation ratio discussion. Our approach is well performed if we fix d and ε beforehand. For future discussions, we may try to apply similar idea to other combinatorial problems and network related algorithm designs.

Acknowledgments This work is supported by NSFC (10671152, 60603003), the Key Project of Chinese Ministry of Education (208161), and NSF (CCF-0514796, CNS-0524429). This work was completed when Dr. Wang Wei and Dr. Zhao Zhang visiting The University of Texas at Dallas.

References

1. Chen, G., Nocetti, F., Gonzalez, J., Stojmenovic, I.: Connectivity based k-hop clustering in wireless networks. In: Proceedings of the 35th Hawaii International Conference on System Sciences (2002)
2. Cheng, X., Huang, X., Li, D., Du, D.: Polynomial-time approximation scheme for minimum CDS in ad hoc wireless networks. Networks **42**, 202–208 (2003)
3. Cheng, X., Ding, M., Du, H., Jia, X.: Virtual backbone construction in multihop ad hoc wireless networks. Wireless Commun. Mobile Comput. **6**, 183–190 (2006)
4. Cokuslu, D., Erciyes, K.: A hierarchical connected dominating set based clustering algorithm for mobile ad hoc networks. In: Proceedings of 15th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS2007, IEEE), pp. 61–66, October 24–26, Istanbul (2007)
5. Gao, X., Li, X., Wu, W.: A Constant-factor approximation for d-hop connected dominating sets in unit disk graphs. IEEE Trans. Mobile Comput. (2008, submitted)
6. Gfeller, B., Vicari, E.: A faster distributed approximation scheme for the connected dominating set problems for growth-bounded graphs. In: Proceedings of the 6th Ad-Hoc, Mobile, and Wireless Networks International Conference (ADHOC-NOW 2007). LNCS 4686, pp. 59–73. Springer, Berlin (2007)

7. Nguyen, T., Huynh, D.: Connected D-hop dominating sets in mobile ad hoc networks. In: Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 1–8 April (2006)
8. Nieberg, T., Hurink, J.: A PTAS for the minimum dominating set problem in unit disk graphs. Third Workshop on Approximation and Online Algorithms: part of ALGO 2005 (WAOA 2005). LNCS 3879, pp. 296–306, Springer, Berlin (2006)
9. Nocetti, F., Gonzalez, J., Stojmenovic, I.: Connectivity based k-hop clustering in wireless networks. Telecommunication Syst. **22**, 1–4, 205–220 (2003)
10. Rieck, M., Pai, S., Dhar, S.: Distributed routing algorithms for multi-hop ad hoc networks using d-hop connected d-dominating sets. Comput. Netw. **47**(6), 785–799 (2005)
11. Shi, Z., Srimani, P.: A new adaptive distributed routing protocol using d-hop dominating sets for mobile ad hoc networks. In: Proceedings of the 2004 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'04), Las Vegas, June 21–24 (2004)
12. Spohn, M., Garcia-Luna-Aceves, J.: Enhancing broadcast operations in ad hoc networks with two-hop connected dominating sets. In: Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04), October 24–27, Fort Lauderdale, Florida, USA (2004)
13. Vuong, T., Huynh, D.: Adapting d-hop dominating sets to topology changes in ad hoc networks. In: Proceedings Ninth International Conference on Computer Communications and Networks, pp. 348–353 (2000)
14. Yau, S., Gao, W.: Multi-hop clustering based on neighborhood benchmark in mobile ad-hoc networks. Mobile Netw. Appl. **12**(5–6), 381–391 (2007)