2014 10th International Conference on Mobile Ad-hoc and Sensor Networks

A New Greedy Algorithm for d-hop Connected Dominating Set

Xianyue Li
School of Mathematics
and Statistics
Lanzhou University
Lanzhou, Gansu, China, 730000
Email: lixianyue@lzu.edu.cn

Xiaofeng Gao
Department of Computer
Science and Engineering
Shanghai Jiao Tong University
Shanghai, China, 200240
Email: gao-xf@cs.sjtu.edu.cn

Chenxia Zhao
School of Mathematics
and Statistics
Lanzhou University
Lanzhou, Gansu, China, 730000
Email: zhaochx@lzu.edu.cn

***Abstract*—In this paper, we consider the problem how to interconnect an obtained d-hop MIS into a d-hop CDS. Firstly, we deal with a simple case $d = 2$ and present a greedy algorithm. Using this method, we can obtain a 2-hop CDS with approximation ratio $\min\{3\beta, 2\beta+2+2H(\beta-1)\}$, where β is the ratio of 2-hop MIS with 2-hop CDS and $H(\cdot)$ is the harmonic function. This ratio is better than the ratio using spanning tree method. Finally, we generalize the algorithm for general case.**

***Keywords*: connected dominating set, multi-hop virtual backbone, approximation algorithm, greedy strategy.**

I. Introduction

Nowadays, wireless networks attract more and more attention from both scientists and engineers. Without the wires, messages in a wireless network are transmitted from one node to another via radio waves between two wireless stations within a transmission range. If two nodes are too far away from each other, they will exchange messages through routing protocols with the help of several intermediate nodes. Due to the characteristics of wireless networks, the lack of physical infrastructure brings the inefficiency and instability for information transmission process.

To improve the performance of communication efficiency, the researchers use connected dominating set (CDS) as the virtual backbone of the wireless networks. Given a graph $G = (V, E)$, a *dominating set* (DS) of G is a sub-vertex-set C such that for any vertex $u \in V \setminus C$, there exists a vertex $v \in C$ with u and v are adjacent in G. Furthermore, if the subgraph $G[C]$ induced by C is connected, we call C is a *connect dominating set* (CDS). To determine a minimum CDS in a general graph, even in a unit disk graph (UDG), is NP-Complete. There are so many works to give approximation algorithms on CDS problems in general graphs and unit disk graphs until now [2], [5], [7], [12].

If we separate the network into many clusters and choose the vertices in CDS as cluster-heads, each node will send message to its local cluster-head and information is exchanged among those cluster-heads through more steady and responsible channels. It makes the whole network more reliable. Easy to see, for a CDS, each cluster is really small, which only includes nodes adjacent with the corresponding cluster-head. In order to enlarge the size of clusters, super cluster-head is presented, which can be at most d-hop away from the nodes within its dominating range. The set of such super cluster-head is called d-hop CDS. As the definition of CDS, given a graph $G = (V, E)$, a d-hop CDS of G is a subset $C \subseteq V$ such that: (i) $G[C]$ is connected, and (ii) for any vertex $u \in V \setminus C$, there exists a vertex $v \in C$ satisfying the distance of u and v at most d, where the distance of two vertices in G is the length of the shortest path interconnecting these two vertices. When $d = 2$, we often call the 2-hop CDS as TCDS. The following figure gives a example of CDS and TCDS in a same graph G.

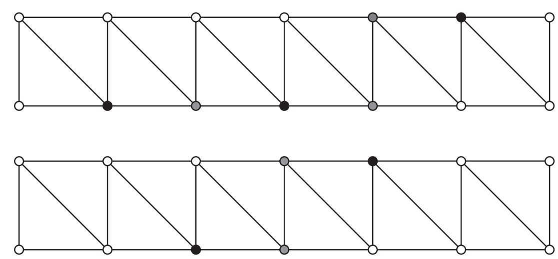

Fig. 1. The top is a CDS and the bottom is a TCDS. The black vertices are DS and 2-hop DS respectively, and the gray vertices are connectors. Both of them are minimum.

Vuong and Huynh [13] proved that to find a minimum d-hop CDS is NP-Complete in general graph, and later Nguyen and Huynh [10] proved that d-hop CDS problem is also NP-Complete in UDGs. Hence, a lot of literatures presented heuristic and approximation algorithms to solve d-hop CDS problem in unit disk graphs and general graphs [3], [11], [15]. Gao et al. [6] gave a two-step approximation algorithm for this problem on UDGs. In that paper, they firstly found a d-hop DS and then interconnected it by spanning tree method in the given graph. Finally, they showed that the size of output is at most $(0.335r^3+1.337r^2+0.585r)opt+(3.338r^3+0.5r^2+0.585r)$, where $r = d + 0.5$ and *opt* is the size of the minimum d-hop CDS in given graph. In [14], Wang et al. gave a PTAS for computing a minimum d-hop DS for growth-bounded graphs (including UDGs). Researchers also studied variants of the problem, such as connected d-hop k-dominating set problem [8], [16], 2-connected d-hop dominating set problem [9], etc.

In this paper, we mainly consider the second phase in two-step method, that is, how to interconnect a given d-hop DS into a d-hop CDS. Since the algorithm of first-step in [6] can also be executed in general graphs, we only consider the interconnection process in general graphs. To illuminate

978-1-4799-7394-1/14 $31.00 © 2014 IEEE
DOI 10.1109/MSN.2014.14

our idea better, we firstly propose an approximation on 2-hop CDS (TCDS) and then generalize it on d-hop CDS problem. Different from the spanning method, we present a greedy algorithm which includes the spanning tree method. We analysis the approximation ratio of the algorithm and show that the ratio is at most $\min\{2\beta, (\beta+2+2H(\beta-1))\}$, where β is the approximation ratio of the first-step and $H(\cdot)$ is the harmonic function.

The rest of this paper is organized as follows. In section 2, we introduce some basic concepts used in this paper. In section 3, we firstly present our greedy algorithm to interconnect a given 2-hop DS into a 2-hop CDS. Then we show the approximation ratio of the algorithm, and generalize this algorithm on d-hop CDS problem. Finally, we give the conclusion and discussion for future works.

II. Preliminary

In this section, we will introduce some useful concepts and results used in our algorithm and analysis. As the same as constructing CDS, researches usually use two-step method to design a d-hop CDS. The first step is finding a d-hop DS, and researchers always use a maximal d-hop independent set as a d-hop DS. Given a graph $G=(V,E)$, a *d-hop independent set* M is a subset of $V(G)$ such that for any two vertices u and v in M, the distance between them is at least d. Furthermore, M is called a *maximal d-hop independent set* (d-hop MIS), if we insert any vertex in $V\backslash M$ into M, then M is not an independent set any more. From the definition, we can see that any d-hop MIS is a d-hop DS.

In [6], Gao et al. presented a distributed algorithm to construct a d-hop MIS on UDGs, and this algorithm can be used in general graphs. Furthermore, the d-hop MIS M obtained by the algorithm has the following property.

If we divide M into two parts M_1 and M_2, the distance between them is d+1.

If a d-hop MIS satisfies this property, we call it a d-hop MIS with property 1. Easy to see, given such a d-hop MIS M, we can connect any bipartition of M by adding at most d new vertices. Repeating this process, we can obtain a d-hop CDS by adding at most $d|M|$ new vertices. This is the idea of spanning tree method in second-step. Hence, if the approximation ratio of the first-step is β, then the approximation ratio of the second-step is $d\beta$.

Given a graph $G=(V,E)$ and a vertex v, let $N^d[v]=\{u|d(u,v)\le d\}$ and $N^d(v)=N^d[v]\backslash\{v\}$. For example, $N(v)$ is vertex set including all vertices adjacent to v, called the neighborhood of v, and $N^2[v]$ collects v, all neighbors of v and all neighbors of neighbors of v. For any vertex subset Y, denoted by $N^d[Y]=\cup_{v\in Y}N^d[v]$ and $N^d(Y)=N^d[Y]\backslash Y$. For any two vertices u and v, and a shortest path P connecting them, let $I_p(u,v)$ be the set of all internal vertices in P. If we omit the subscript P, it means collecting all internal vertices in any arbitrary shortest path.

Any other notations and terminologies on graph theory and approximation algorithm does not mentioned here can refer to textbooks [1], [4].

III. Main Results

In this section, we will present a greedy algorithms for interconnecting the d-hop MIS into a d-hop CDS. To better illuminate our idea, we firstly consider the special case when $d=2$, and then study the general case.

A. Greedy algorithm for TCDS

At the beginning, we give some useful notations. Let M be a 2-MIS of G with property 1. For any vertex v of G and any subset M_0 of M, denoted by $C_{v,M_0}=N^2(v)\cap(M-M_0)$ and $C'_{v,M_0}=(\cup_{u\in C_{v,M_0}}I(v,u))\bigcup I(v,M_0)\bigcup\{v\}$, which collects vertex v, all internal vertices in an arbitrary shortest path connecting v and M_0, and all internal vertices in some shortest paths connecting v and all vertices in C_{v,M_0}.

The pseudo-code of our algorithm as follows.

Algorithm 1 A greedy algorithm for TCDS

Input: A 2-hop MIS(DS) M with property 1;
Output: A 2-hop CDS D;

1: **Initialize**: choose an arbitrary vertex u from M, set $M_0:=\{u\}$ and $C:=\emptyset$;
2: **while** $M_0\neq M$ **do**
3: find a vertex $v\in N^2(M_0)$ with C_{v,M_0} as large as possible; (**Note**: choosing v in $N(M_0)$ possibly, if we can)
4: set $M_0:=M_0\cup C_{v,M_0}$ and $C:=C\cup C'_{v,M_0}$;
5: **end while**
6: **return** $D=C\cup M_0$;

Theorem 1. *The algorithm can terminate in finite steps and the output is a 2-hop CDS.*

Proof: We consider the induced graph $G[C\cup M_0]$ in each stage of the algorithm. Since C'_{v,M_0} contains vertices which interconnect current M_0 and all vertices of C_{v,M_0}, if the $G[C\cup M_0]$ is connected, the updated $C\cup M_0$ induces a connected subgraph. In initial stage, $G[C\cup M_0]=G[u]$ is connected, so the output $D=C\cup M_0=C\cup M$ is a 2-hop CDS.

To show the algorithm terminates in finite steps, we only to prove that if $M_0\neq M$, there always exists a vertex v with $|C_{v,M_0}|\ge 1$. Since the input 2-MIS M satisfying property 1, we have $d(M_0,M-M_0)=3$. Hence, any internal vertex v in any shortest path connecting M_0 and $M-M_0$ is a candidate vertex and $|C_{v,M_0}|\ge 1$. ■

B. Theoretical Analysis

Let β be the ratio of 2-hop MIS and TCDS. The following theorem gives the approximation ratio of the algorithm.

Theorem 2. *When the algorithm terminates, $|C|\le\min\{2\beta,\beta+2+2H(\beta-1)\}\cdot opt$, where opt is the size of minimum TCDS and $H(\cdot)$ is the harmonic function.*

Proof: To distinct M_0 in the algorithm, we use M_0^i to denote it in stage i and assume that the algorithm terminates after t stages. At the beginning of the proof, we give each vertex x in M a value $w(x)$ as follows. For the initial vertex u, let $w(u)=0$ and for any other vertex $x\in C_{v,M_0^i}$, let

$w(x)=\frac{|C'_{v,M_0^i}|}{|C_{v,M_0^i}|}$. Based on this definition, we have $|C'_{v,M_0^i}| = w(x)\cdot|C_{v,M_0^i}| = \sum\limits_{x\in C_{v,M_0^i}} w(x)$ and

$$\begin{aligned}
|C| &= |C'_{v,M_0^1}\cup C'_{v,M_0^2}\cup\cdots\cup C'_{v,M_0^t}| \\
&\leq 0+|C'_{v,M_0^1}|+|C'_{v,M_0^2}|+\cdots+|C'_{v,M_0^t}| \\
&= w(u)+\sum_{x\in C_{v,M_0^1}} w(x)+\cdots+\sum_{x\in C_{v,M_0^t}} w(x) \\
&= \sum_{x\in M} w(x).
\end{aligned}$$

Hence, we can use the sum of value of all vertices in M to estimate $|C|$. By the choosing method of C'_{v,M_0}, we can see that $|C'_{v,M_0^i}|\leq 2+|C_{v,M_0^i}|$, that is,

$$w(x)\leq\frac{2+|C_{v,M_0^i}|}{|C'_{v,M_0^i}|}=1+\frac{2}{|C_{v,M_0^i}|}. \tag{1}$$

Easy to see, $w(x)\leq 3$ since $|C_{v,M_0^i}|\geq 1$. But if $|C_{v,M_0^i}|=1$, based on the note in line 3, we choose the vertex v in $N(M_0)$ and $|C'_{v,M_0^i}|=2$ in this case. It implies that $w(x)\leq 2$ and

$$|C|\leq\sum_{x\in M} w(x)\leq 2|M|\leq 2\beta\cdot opt. \tag{2}$$

Next, we will give a partition of $V(G)$. Let $S=\{s_1,s_2,\ldots,s_m\}$ be a minimum 2-hop CDS in G and $V'=V\backslash S$. Let S_i be union of s_i and the set of vertices in V' which can be 2-dominated by s_i only, but cannot be 2-dominated by any s_j with $1\leq j\leq i-1$, that is,

$$\begin{aligned}
S_1 &= s_1\cup(N^2[s_1]\cap V'); \\
S_2 &= s_2\cup((N^2[s_2]\cap V')\backslash S_1)); \\
S_3 &= s_3\cup((N^2[s_3]\cap V')\backslash(S_1\cup S_2)); \\
&\vdots \\
S_i &= s_i\cup((N^2[s_i]\cap V')\backslash(\bigcup_{j=1}^{i-1} S_j)); \\
&\vdots
\end{aligned}$$

Since S is a TCDS of G, $\{S_1,S_2,\ldots,S_m\}$ is a partition of $V(G)$ and combined with Eqn. 2, we have

$$|C|\leq\sum_{x\in M} w(x)=\sum_{i=1}^{m}(\sum_{x\in(M\cap S_i)} w(x)) \tag{3}$$

Final, for each s_i, we will obtain an upper bound of $\sum\limits_{x\in(M\cap S_i)} w(x)$. Conveniently, we use M^j to denote the set of vertices which are not containing in the M_0 at the end of stage j, that is, $M^j=M\backslash M_0^j$. Easy to see, $M^0=M\backslash\{u\}$, $M^t=\emptyset$ and $|M^0|>|M^1|>\cdots>|M^t|=0$. Let $a_j=|(M\cap S_i)\cap M^j|=|M^j\cap S_i|$, the number of vertices in $M\cap S_i$ not belonging to M_0 at the end of stage j. Clearly, $a_0\geq a_1\geq a_2\geq\cdots\geq a_t=0$. Now, we will consider the following conditions based on the a_0.

Case 1. $a_0=0$**.** It implies that $M\cap S_i=\emptyset$ and

$$\sum_{x\in(M\cap S_i)} w(x)=0.$$

Case 2. $a_0=1$**.** It implies that $M\cap S_i=\{x\}$ and

$$\sum_{x\in(M\cap S_i)} w(x)=w(x)\leq 2\leq a_0+2+2H(a_0-1).$$

Case 3. $a_0\geq 2$**.** It implies that $s_i\notin M$. Since any vertex of M is valued when it is just added into M_0 in the algorithm, we can only consider the stages j with $a_{j-1}>a_j$. Let $j_1,j_2,\ldots,j_k$ be such stages and $a_0>a_{j_1}>\cdots>a_{j_k}=0$. In j_1 stage, there are $a_0-a_{j_1}$ vertices in S_i added into M_0, so at least $a_0-a_{j_1}$ vertices added into M_0 of G. It implies that $|C_{v,M_0}|\geq a_0-a_{j_1}$ and $w(x)\leq 1+\frac{2}{|C_{v,M_0^i}|}\leq 1+\frac{2}{a_0-a_{j_1}}$. Thus, the total value of vertices added into M_0 in $S_i\cap M$ at this stage is at most $(a_0-a_{j_1})\cdot(1+\frac{2}{a_0-a_{j_1}})$. In any other stage j_q with $q\geq 2$, there exists some vertex v in $S_i\cap M$ already belonging to M_0 because of $a_{j_{q-1}}<a_0$. Hence, s_i is a candidate vertex of this stage and $|C_{v,M_0}|\geq a_{j_{q-1}}$ now. Thus, we have $w(x)\leq 1+\frac{2}{a_{j_{q-1}}}$ and the total value of vertices added into M_0 in $S_i\cap M$ at this stage is at most $(a_{j_{q-1}}-a_{j_q})\cdot(1+\frac{2}{a_{j_{q-1}}})$. As above, we can obtain that

$$\begin{aligned}
\sum_{x\in(M\cap S_i)} w(x) &\leq (a_0-a_{j_1})\cdot(1+\frac{2}{a_0-a_{j_1}}) \\
&\quad+\sum_{q=2}^{k}(a_{j_{q-1}}-a_{j_q})\cdot(1+\frac{2}{a_{j_{q-1}}}) \\
&= (a_0-a_{j_1})+\sum_{q=2}^{k}(a_{j_{q-1}}-a_{j_q}) \\
&\quad+2(1+\sum_{q=2}^{k}(a_{j_{q-1}}-a_{j_q})\cdot\frac{1}{a_{j_{q-1}}}) \\
&\leq (a_0-a_{j_k})+2 \\
&\quad+2(\sum_{q=2}^{k}(\frac{1}{a_{j_{q-1}}}+\frac{1}{a_{j_{q-1}}-1}+\cdots+\frac{1}{a_{j_q}+1}) \\
&= a_0+2(1+H(a_{j_1})) \\
&\leq a_0+2+2H(a_0-1)
\end{aligned}$$

To distinct each s_i, let $a_0^i=|(M\cap S_i)|$. Based on the above 3 cases and combined with Eqn. 3, we have

$$\begin{aligned}
|C| &\leq \sum_{i=1}^{m}(\sum_{x\in(M\cap S_i)} w(x)) \\
&\leq \sum_{i=1}^{m}(a_0^i+2+2H(a_0^i-1)) \\
&= \sum_{i=1}^{m} a_0^i+2m+\sum_{i=1}^{m}2H(a_0^i-1)) \\
&\leq |M|+2m+2m\cdot H(\max_{1\leq i\leq m}\{a_0^i\}-1) \\
&\leq \beta\cdot opt+2\cdot opt+2H(\beta-1)\cdot opt \\
&= (\beta+2+2H(\beta-1))\cdot opt
\end{aligned}$$

Combined with Eqn. 2, the proof is done. ■

C. Generalization

Our algorithm can be easily generalized to d-hop CDS problem, if we replace 2 by d in the algorithm and the definition of C_{v,M_0}. Using the same method to analysis the generalized algorithm, we can show that the size of output is at most $\min\{d\beta, (d-1)\beta + d + dH(\beta - 1)\} \cdot opt$.

IV. Conclusion

In this paper, we firstly present a greedy algorithm for interconnecting a given 2-hop MIS into a 2-hop CDS. And then, we show that the approximation ratio of the algorithm is $\min\{2\beta, \beta + 2 + 2H(\beta - 1)\}$, where β be the ratio of 2-MIS and TCDS and $H(\cdot)$ is the harmonic function, which is better than that of spanning tree method. Final, we generalize the algorithm into general case.

Actually, there are a lot of future works on this problem. For example, recalling the algorithm, the longest distance between M_0 and C_{v,M_0} is $2d$. So we can construct the d-hop MIS with nearest distance $2d$ instead of d+1 and if we do this, the size of d-hop MIS in the first-step may be reduced. On the other side, we can search the vertices in $N^{d/2}(M_0)$ and compare the size of union of these $\frac{d+1}{2}$-neighborhood and $M \setminus M_0$. To do this can decrease the number of connectors, but it is may difficult to analysis the approximation ratio.

Acknowledgment

This work is partly supported by NSFC (No. 11201208 and 61202024), and the Fundamental Research Funds for the Central Universities (No. lzujbky-2011-44).

References

[1] J.A. Bondy and U.S.R. Murty, *Graph Theory*, Springer, New York, 2007.

[2] X.Z. Cheng, X.Huang, D.Y. Li, W.-L. Wu and D.-Z. Du, Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks, *Networks*, 42 202-208, 2003.

[3] D. Cokuslu and K. Erciyes, A Hierarchical Connected Dominating Set Based Clustering Algoithm for Mobile Ad hoc Networks, *Proceedings of 15th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* 61-66, 2007.

[4] D.-Z. Du, K.-I. Ko and X. Hu, *Design and Analysis of Approximation Algorithms*, Springer-Verlag, New York, 2011.

[5] X. Gao, Y. Wang, X. Li and W. Wu, Analysis on theoretical bounds for approximating dominating set problems, *Discrete Mathematics, Algorithms and Applications*, 1 71-84, 2009.

[6] X. Gao, W. Wu, X. Zhang and X. Li, A Constant-Factor Approximation for d-Hop Connected Dominating Sets in Unit Disk Graph, *International Journal of Sensor Networks*, 12(3) 125-136, 2012.

[7] M. Guha and S. Khuller, Approximation algorithms for connected dominating set, *Algorithmica*, 20 374-387, 1998.

[8] D. Li, L. Liu and H. Yang, Minimum connected r-hop k-dominating set in wireles networks, *Discrete Mathematics, Algorithms and Applications*, 1 45-58, 2009.

[9] X. Li and Z. Zhang, Two algorithms for minimum 2-connected r-hop dominating set, *Information Processing Letters*, 110 986-991, 2010.

[10] T.N. Nguyen and D.T. Huynh, Connected D-Hop Dominating Sets in Mobile Ad Hoc Networks, *Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 1-8, 2006.

[11] M.Q. Rieck, S. Pai and S. Dhar, Distributed Routing Algorithms for Multi-hop Ad Hoc Networks using d-Hop Connected d-Dominating Sets, *Computer networks*, 47(6) 785-799, 2005.

[12] L. Ruan, H.W. Du, X.H. Jia, W.L. Wu, Y.S. Li and K. Ko, A greedy approximation for minimum connected dominating sets, *Theoretical Computer Science*, 329 325-330, 2004.

[13] T.HP. Vuong and D.T. Huynh, Adapting D-hop Dominating Sets to Topology Changes in Ad Hoc Networks, *Proceedings Ninth International Conference on Computer Communications and Networks*, 348-353, 2000.

[14] W. Wang, D. Kim, N. Sohaee, C. Ma and W. Wu, A PTAS for minimum d-hop underwater sink placement problem in 2-D underwater sensor networks, *Discrete Mathematics, Algorithms and Applications*, 1 283-290, 2009.

[15] S. Yau and W. Gao, Multi-hop Clustering Based on Neighborhood Benchmark in Mobile Ad-hoc Networks, *Mobile Networks and Applications*, 12 381-391, 2007.

[16] Z. Zhang, Q. Liu and D. Li, Two Algorithms for Connected r-hop k-dominating Set, Discrete Mathematics, *Algorithms and Applications*, 1(4) 485-498, 2009.