Algorithms for Minimum m-Connected k-Dominating Set Problem*

Weiping Shang[1,2], Frances Yao[2], Pengjun Wan[3], and Xiaodong Hu[1]

[1] Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China
[2] Department of Computer Science, City University of Hong Kong
[3] Department of Computer Science, Illinois Institute of Technology, Chicago, USA

Abstract. In wireless sensor networks, virtual backbone has been proposed as the routing infrastructure to alleviate the broadcasting storm problem and perform some other tasks such as area monitoring. Previous work in this area has mainly focused on how to set up a small virtual backbone for high efficiency, which is modelled as the minimum Connected Dominating Set (CDS) problem. In this paper we consider how to establish a small virtual backbone to balance efficiency and fault tolerance. This problem can be formalized as the minimum m-connected k-dominating set problem, which is a general version of minimum CDS problem with $m = 1$ and $k = 1$. In this paper we will propose some approximation algorithms for this problem that beat the current best performance ratios.

Keywords: Connected dominating set, approximation algorithm, k-vertex connectivity, wireless sensor networks.

1 Introduction

A Wireless Sensor Network (WSN) consists of wireless nodes (transceivers) without any underlying physical infrastructure. In order to enable data transmission in such networks, all the wireless nodes need to frequently flooding control messages thus causing a lot of redundancy, contentions and collisions. To support various network functions such as multi-hop communication and area monitoring, some wireless nodes are selected to form a *virtual backbone*. Virtual backbone has been proposed as the routing infrastructure of WSNs. In many existing schemes (e.g., [1]) virtual backbone nodes form a Connected Dominating Set (CDS) of the WSN. With virtual backbones, routing messages are only exchanged between the backbone nodes, instead of being broadcasted to all the nodes. Prior work (e.g., [8]) has demonstrated that virtual backbones could dramatically reduce routing overhead.

In WSNs, a node may fail due to accidental damage or energy depletion and a wireless link may fade away during node movement. Thus it is desirable to

* This work was supported in part by the Research Grants Council of Hong Kong under Grant No. CityU 1165/04E, the National Natural Science Foundation of China under Grant No. 70221001 and 10531070.

A. Dress, Y. Xu, and B. Zhu (Eds.): COCOA 2007, LNCS 4616, pp. 182–190, 2007.

have several sensors monitor the same target, and let each sensor report via different routes to avoid losing an important event. Hence, how to construct a fault tolerant virtual backbone that continues to function when some nodes or links break down is an important research problem.

In this paper we assume as usual that all nodes have the same transmission range (scaled to 1). Under such an assumption, a WSN can be modelled as a Unit Disk Graph (UDG) that consists of all nodes in the WSN and there exists an edge between two nodes if the distance between them is at most 1. Fault tolerant virtual backbone problem can be formalized as a combinatorial optimization problem: Given a UDG $G = (V, E)$ and two nonnegative integers m and k, find a subset of nodes $S \subseteq V$ of minimum size that satisfies: i) each node u in $V \setminus S$ is *dominated* by at least k nodes in S, ii) S is *m-connected* (there are at least m disjoint paths between each pair of nodes in S). Every node in S is called a *backbone node* and every set S satisfying (i-ii) is called *m-connected k-dominating set* ((m, k)-CDS), and the problem is called *minimum m-connected k-dominating set* problem.

In this paper, we will first study the minimum m-connected k-dominating set problem for $m = 1, 2$, which is important for fault tolerant virtual backbone problem in WSNs. (When $m = 1$ and $k = 1$ the problem is reduced to well known minimum connected dominating set problem.) We propose three centralized approximation algorithms to construct k-dominating set and m-connected k-dominating sets for $m = 1, 2$. Our performance analysis show that the algorithms have small approximation ratio improving the current best result for small k. Then for $3 \leq m \leq k$, we discuss the relation between (m, k)-CDS and (m, m)-CDS. The remainder of this paper is organized as follows: In Section 2 and 3 we first give some definitions and then present some related works. In Section 4 we present our algorithms with theoretical analysis on guaranteed performances. In Section 5 we conclude the paper.

2 Preliminaries

Let G be a graph with vertex-set $V(G)$ and edge-set $E(G)$. For any vertex $v \in V$, the neighborhood of v is defined by $N(v) \equiv \{u \in V(G) : uv \in E(G)\}$ and the closed neighborhood of v is defined by $N[v] \equiv \{u \in V(G) : uv \in E(G)\} \cup \{v\}$. The minimum degree of vertices in $V(G)$ is denoted by $\delta(G)$.

A subset $U \subseteq V$ is called an *independent set* (IS) of G if all vertices in U are pairwise non-adjacent, and it is further called a *maximal independent set* (MIS) if each vertex $V \setminus U$ is adjacent to at least one vertex in U.

A *dominating set* (DS) of a graph $G = (V, E)$ is a subset $S \subseteq V$ such that each vertex in $V \setminus S$ is adjacent to at least one vertex in S. A DS is called a *connected dominating set* (CDS) if it also induces a connected subgraph. A *k-dominating set* (k-DS) $S \subseteq V$ of G is a set of vertices such that each vertex $u \in V$ is either in S or has at least k neighbors in S.

A *cut-vertex* of a connected graph G is a vertex v such that the graph $G \setminus \{v\}$ is disconnected. A *block* is a maximal connected subgraph having no cut-vertex (so a graph is a block if and only if it is either 2-connected or equal to K_1 or K_2). The block-cut-vertex graph of G is a graph H where $V(H)$ consists of all cut-vertices of G and all blocks of G, with a cut-vertex v adjacent to a block G_0 if v is a vertex of G_0. The block-cut-vertex graph is always a forest. A *2-connected graph* is a graph without cut-vertices. Clearly a block with more than three nodes is a 2-connected component. A *leaf block* of a connected graph is a subgraph of which is a block with only one cut-vertex.

3 Related Work

Lots of efforts have been made to design approximation algorithms for minimum connected dominating set problem. Wan et al. [10] proposed a two-phase distributed algorithm for the problem in UDGs that has a constant approximation performance ratio of 8. The algorithm first constructs a spanning tree, and then at the first phase, each node in a tree is examined to find a *Maximal Independent Set* (MIS) and all the nodes in the MIS are colored black. At the second phase, more nodes are added (color blue) to connect those black nodes. Recently, Li et al. [6] proposed another two-phase distributed algorithm with a better approximation ratio of $(4.8 + ln5)$. As in [10], at the first phase, an MIS is computed. At the second phase, a Steiner tree algorithm is used to connect nodes in the MIS. The Steiner tree algorithm is based on the property that any node in UDG is adjacent to at most 5 independent nodes.

In [3], Dai et al address the problem of constructing k-connected k-dominating virtual backbone which is k-connected and each node not in the backbone is dominated by at least k nodes in the backbone. They propose three localized algorithms. Two algorithms, k-gossip algorithm and color based (k, k)-CDS algorithm, are probabilistic. In k-Gossip algorithm, each node decides its own backbone status with a probability based on the network size, deploying area size, transmission range, and k. Color based (k, k)-CDS algorithm proposes that each node randomly selects one of the k colors such that the network is divided into k-disjoint subsets based on node colors. For each subset of nodes, a CDS is constructed and (k, k)-CDS is the union of k CDS's. The deterministic algorithm, k-Coverage condition, only works in very dense network and no upper bound on the size of resultant backbone is analyzed.

Recently, Wang et al. [11] proposed a 64-approximation algorithm for the minimum $(2, 1)$-CDS problem. The basic idea of this centralized algorithm is as follows: i) Construct a small-sized CDS as a starting point of the backbone; ii) iteratively augment the backbone by adding new nodes to connect a leaf block in the backbone to other block (or blocks); iii) the augmentation process stops when all backbone nodes are in the same block, i.e., the backbone nodes are 2-connected. The augmentation process stops in at most $|CDS| - 1$ steps and each step at most 8 nodes are added.

Most recently, in work [7] we proposed three centralized approximation algorithms to construct k-tuple dominating set and m-connected k-tuple dominating sets for $m = 1, 2$, respectively.

4 Approximation Algorithms

We first prove the following lemma, which will be used in our performance analysis of proposed algorithms.

Lemma 1. *Let $G = (V, E)$ be a unit disk graph and k a constant such that $\delta(G) \geq k - 1$. Let D_k^* be a minimum k-dominating set of G and S a maximal independent set of G. Then $|S| \leq \max\{\frac{5}{k}, 1\}|D_k^*|$.*

Proof Let $S_0 = S \bigcap D_k^*$, $X = S \setminus S_0$ and $Y = D_k^* \setminus S_0$. It is clearly that X and Y are two disjoint subsets. For all $u \in X$, let $c_u = |N(u) \bigcap Y|$. As D_k^* is a k-dominating set of G, $c_u \geq k$ for each $u \in X$ and we have: $\sum_{u \in X} c_u \geq k|X|$. For all $v \in Y$, let $d_v = |N(v) \bigcap X|$. As G is a unit disk graph, for all $v \in Y$ there are at most 5 independent vertices in its neighborhood and $d_v \leq 5$. We have: $5|Y| \geq \sum_{v \in Y} d_v$. For $\sum_{u \in X} c_u = |\{uv \in E : u \in X, v \in Y\}| = \sum_{v \in Y} d_v$, we have $|X| \leq \frac{5}{k}|Y|$. Hence, $|S| = |X| + |S_0| \leq \frac{5}{k}|D_k^* \setminus S_0| + |S_0| \leq \max\{\frac{5}{k}, 1\}|D_k^*|$, which proves the lemma. □

Corollary 1. *Let $G = (V, E)$ be a unit disk graph and k a constant such that $\delta(G) \geq k-1$. Let D_k^* be a minimum k-dominating set of G and S an independent set of G satisfying that $S \bigcap D_k^* = \emptyset$. Then $|S| \leq \frac{5}{k}|D_k^*|$.*

4.1 Algorithm for Computing $(1, k)$-CDS

The basic idea of our algorithm for the minimum $(1, k)$-CDS problem is as follows: First choosing a CDS and then sequentially choosing an MIS $k - 1$ times such that all vertices in $V \setminus D^c$ are k-dominated by set D^c. The algorithm is more formally presented as follows.

Algorithm A. for computing $(1, k)$-CDS

1. Choose an MIS I_1 of G and a set C such that $I_1 \cup C$ is a CDS (refer to [10])
2. **for** $i := 2$ **to** k
3. Construct an MIS I_i in $G \setminus I_1 \cup \cdots \cup I_{i-1}$
4. **end for**
5. $D^c := I_1 \cup \cdots \cup I_k \cup C$
6. **return** D^c

Theorem 1. *Algorithm A returns a solution that is a $(5 + \frac{5}{k})$-approximate solution to the minimum connected k-dominating set problem for $k \leq 5$ and 7-approximate solution for $k > 5$.*

Proof: Suppose that Algorithm A, given graph $G = (V, E)$ and a natural number $k \geq 1$, returns $D^c = I_1 \cup \cdots \cup I_k \bigcup C$. Let D_k^* be a minimum k-dominating set of G. We will show that D is a connected k-dominating set of G. For all $u \in G \setminus D^c$, at the i-th iteration, u is not in I_i and thus it is dominated by one vertex of I_i. At the end, u is dominated by at least k different vertices of $I_1 \cup \cdots \cup I_k$. By the first step of Algorithm A, $C \cup I_1$ is a CDS and thus $I_1 \cup \cdots \cup I_k \bigcup C$ is connected. So, D is a connected k-dominating set of G.

Let $S_i = I_i \bigcap D_k^*$ for $i = 1, 2, \cdots, k$. By the rule of Algorithm A, we have each $I_i \setminus S_i$ is an independent set and $(I_i \setminus S_i) \bigcap D_k^* = \emptyset$. Thus it follows from Corollary 1 that $|I_i \setminus S_i| \leq \frac{5}{k}|D_k^* \setminus S_i|$. Let us prove now the approximation ratio.

$$\begin{aligned}
|I_1 \cup \cdots \cup I_k| &= \sum_{i=1}^{k} |S_i| + \sum_{i=1}^{k} |I_i \setminus S_i| \\
&\leq \sum_{i=1}^{k} |S_i| + \sum_{i=1}^{k} \frac{5}{k}|D_k^* \setminus S_i| \\
&= (1 - \frac{5}{k}) \sum_{i=1}^{k} |S_i| + 5|D_k^*|.
\end{aligned}$$

And $\sum_{i=1}^{k} |S_i| \leq |D_k^*|$. Hence we have $|I_1 \cup \cdots \cup I_k| \leq 5|D_k^*|$ for $k \leq 5$ and $|I_1 \cup \cdots \cup I_k| \leq 6|D_k^*|$ for $k > 5$.

In the end, let C be the set constructed from the first step of Algorithm A. By using the argument for the proof of Lemma 10 in [10], we can deduce $|C| \leq |I_1|$. Hence it follows from Lemma 1 that $|C| \leq \max\{\frac{5}{k}, 1\}|D_k^*|$, and the size of connected k-dominating set D is bounded by $(5 + \frac{5}{k})|D_k^*|$ for $k \leq 5$ and $7|D_k^*|$ for $k > 5$. The size of the optimal solution of connected k-dominating set is at least $|D_k^*|$. The proof is then finished. □

4.2 Algorithm for Computing $(2, k)$-CDS

The basic idea of our algorithm for the minimum $(2, k)$-CDS problem with $k \geq 2$ is similar to the method proposed in [11]. It essentially consists of following four steps:

Step 1. Apply Algorithm A to construct a connected k-dominating set D.

Step 2. Compute all the blocks in D by computing the 2-connected components through the depth first search.

Step 3. Produce the shortest path in the original graph such that it can connect a leaf block in D with other part of D but does not contain any vertices in D except the two endpoints. Then add all intermediate vertices in this path to D.

Step 4. Repeat Step 2 and Step 3 until D is 2-connected.

In Step 2, we can apply the standard algorithm proposed in [9] to compute all blocks in D, denote the number of blocks in D by ComputeBlock(D). The algorithm is more formally presented as follows:

Algorithm B. for computing a 2-connected k-dominating set ($k \geq 2$)

---

1. Choose a connected k-dominating set D^c using Algorithm A
2. $D := D^c$ and B:= ComputeBlocks(D)
3. **while** $B > 1$ **do**
4.   Choose a leaf block L
5.   **for** vertex $v \in L$ not a cut-vertex **do**
6.   **for** vertex $u \in V \setminus L$ **do**
7.   Construct G' from G by deleting all nodes in D except u and v
8. P_{uv} :=shortestPath($G'; v, u$) and $P := P \cup P_{uv}$
9.   **end-for**
10.   **end-for**
11. P_{ij}:= the shortest path in P
12. $D := D\cup$ the intermediate vertices in P_{ij}
13.   ComputeBlocks(D)
14. **end-while**
15. **return** D

Lemma 2. *For $k \geq 2$, at most two new vertices are added into D at each augmenting step.*

Proof Suppose that L is a leaf block of D and w is the cut-vertex. Consider two vertices u and v in D with $u \in L \setminus \{w\}$ and $v \in V \setminus L$, let P_{uv} be the shortest path that connects u and v. We claim that P_{uv} has at most two intermediate vertices. Suppose, by contradiction, that P_{uv} contains $u, x_1, x_2, ..., x_l, v$, where $l \geq 3$. Since each vertex x_i has at least 2 neighbors in D and $N(x_i) \cap D \subseteq L$ or $N(x_i) \cap D \subseteq (V \setminus L) \cup \{w\}$, $N(x_1) \cap D \subseteq L$. If $N(x_2) \cap D \subseteq L$, x_2 must have a neighbor s in $L \setminus \{w\}$, then the path between sv has a shorter distance than P_{uv}. Otherwise $N(x_2) \bigcap D \subseteq (V \setminus L) \cup \{w\}$, x_2 must have a neighbor s in $V \setminus L$, then the path between us has a shorter distance than P_{uv}. Which contradicts that P_{uv} has the shortest distance. □

Lemma 3. *The number of cut-vertices in the connected k-dominating set D^c by Algorithm A is no bigger than the number of connected dominating sets in $I_1 \cup C$ chosen in Step 1 of Algorithm A.*

Proof Let $S = I_1 \cup C$ be the connected domination set. We will show that no vertex in $D^c \setminus S$ is a cut-vertex. For any two vertices $u, v \in S$, there is a path P_{uv} between them that contains only vertices in S. Since any vertex in $D^c \setminus S$ is dominated by at least one vertex in S, Hence, for any two vertices $u, v \in D^c$, there is a path P_{uv} between them that contains only vertices in $S \bigcup \{u, v\}$. Hence, any vertex in $D^c \setminus S$ is not a cut-vertex. □

Theorem 2. *Algorithm B returns a* $(5+\frac{25}{k})$*-approximate solution to the minimum 2-connected k-dominating set problem for* $2 \le k \le 5$ *and 11-approximate solution for* $k > 5$.

Proof Let D_k^* and D_{opt} be the optimal k-dominating set and 2-connected k-dominating set, respectively. It is clearly that $|D_k^*| \le |D_{opt}|$. After S is constructed, by Lemmas 2-3, the algorithm terminates in at most $|C| + |I_1|$ steps, and in each step at most two vertices are added. Since $|C| + |I_1| \le 2|I_1| \le 2\max\{\frac{5}{k}, 1\}|D_k^*|$, we have $|D| \le |D^c| + 4\max\{\frac{5}{k}, 1\}|D_k^*|$. It follows from Theorem 1 that $|D^c| \le (5+\frac{5}{k})|D_k^*|$ for $k \le 5$ and $|D^c| \le 7|D_k^*|$ for $k > 5$. Hence we obtain $|D| \le (5+\frac{25}{k})|D_{opt}|$ for $2 \le k \le 5$ and $|D| \le 11|D_{opt}|$ for $k > 5$. □

4.3 Algorithm for Computing $(2,1)$-CDS

The main idea of our algorithm is as follows: First, construct a connected dominating set C using the algorithm in [6], and then construct a maximal independent set D in $G \setminus C$, in the end make $C \cup D$ to be 2-connected by adding some new vertices to it.

Algorithm C. for computing 2-connected dominating set

---

1. Produce a connected dominating set C of G using the algorithm in [6].
2. Construct a maximal independent set D in $G \setminus C$
3. $S := C \cup D$
4. Augment S using Steps 2-14 of Algorithm B

Theorem 3. *Algorithm C returns a 2-connected dominating set whose size is at most* $(18.2+3\ln 5)|D_2^*|+4.8$*, where* $|D_2^*|$ *is the size of the optimal 2-connected dominating set.*

Proof Let D_1^* and D_2^* be the optimal $(1,1)$-CDS and $(2,1)$-CDS, respectively. It is clear that $|D_1^*| \le |D_2^*|$. After C and D is constructed, which are a connected dominating set of G and a dominating set of $G \setminus C$, respectively, each vertex in $V \setminus S$ is dominated by at least two vertices in S. Thus, Lemmas 2-3 also hold true for Algorithm C. Thus it follows from Lemmas 2-3 that at most $|C|$ steps are needed before the algorithm terminates, and at each step at most two vertices are added. Hence, we obtain $|S| \le 3|C|+|D|$. Using the same argument for Theorem 1 in [6,12], we could show $|C| \le (4.8+\ln 5)|D_1^*| + 1.2$ and $|D| \le 3.8|D_1^*| + 1.2$ respectively. Thus we obtain $|S| \le (18.2+3\ln 5)|D_2^*| + 4.8$. □

Observe that $(18.2 + 3\ln 5) < 23.03$. So Algorithm C has a better guaranteed performance than the 64-approximation algorithm in [11] for the same problem (when the size of the optimal 2-connected dominating set is not very big).

4.4 (m, k)-CDS for $3 \leq m \leq k$

Let $A_{(m,m)}$ be an α-approximation algorithm for the (m, m)-CDS problem. The basic idea of algorithm $A_{(m,k)}$ for the minimum (m, k)-CDS problem is as follows: First choosing a (m, m)-CDS and then sequentially choosing an MIS $k-m$ times. The algorithm is more formally presented as follows.

Algorithm $A_{(m,k)}$**.** for computing (m, k)-CDS

1. Choose an (m, m)-CDS S of G using algorithm $A_{(m,m)}$
2. **for** $i := 1$ **to** $k - m$
3. Construct an MIS I_i in $G \setminus S \cup I_1 \cup \cdots \cup I_{i-1}$
4. $D := I_1 \cup \cdots \cup I_{k-m} \cup S$
5. **return** D

Theorem 4. *If there exists an α-approximation algorithm for the (m, m)-CDS problem, then there exists a $(\alpha + 6)$-approximation algorithm for the (m, k)-CDS problem, where $k > m$.*

Proof: We first show that D is a (m, k)-CDS of G. For all $u \in G \setminus D$, u is not in S and thus it is dominated by at least m vertices of S. And at the i-th iteration, u is not in I_i and thus it is dominated by one vertex of I_i for $i = 1, ..., k - m$. At the end, u is dominated by at least k different vertices of D. Now we show that D is m-connected, suppose there exist $m - 1$ vertices in D such that the induced subgraph D is disconnected by removing the $m - 1$ vertices. Let X be the vertex set. For S is a (m, m)-CDS, $S \setminus X$ is a connected dominating set. So, $D \setminus X$ is connected, a contraction. Hence, D is a (m, k)-CDS of G.

Let D^* be the optimal solution of (m, k)-CDS. It is clearly that $|S| \leq \alpha|D^*|$, and $|I_1 \cup \cdots \cup I_{k-m}| \leq 6|D^*|$ by similar argument of Theorem 1. This gives a $(\alpha + 6)$-approximation algorithm for the (m, k)-CDS problem, where $k > m$. The proof is then finished. □

5 Conclusion

In this paper we have proposed centralized approximation algorithms for the minimum m-connected k-dominating set problem for $m = 1, 2$. Although the approximation performance ratios of Algorithms A and B are dependent on k, they are very small when k is not very big, that, in fact, is the case of virtual backbone construction in wireless sensor networks. For $3 \leq m \leq k$, we discuss the relation between (m, k)-CDS and (m, m)-CDS. Our future work is to extend our study to the more general case of $m \geq 3$, and design distributed and localized algorithms for minimum m-connected k-dominating set problem.

References

1. Alzoubi, K.M., Wan, P.-J., Frieder, O.: Distributed heuristics for connected dominating sets in wireless ad hoc networks. Journal of Communications and Networks 4(1), 22–29 (2002)
2. Bredin, J.L., Demaine, E.D., Hajiaghayi, M., Rus, D.: Deploying sensor networks with guaranteed capacity and fault tolerance. In: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 309–319. ACM Press, New York (2005)
3. Dai, F., Wu, J.: On constructing k-connected k-dominating set in wireless networks. IEEE International Parallel and Distributed Processing Symposium. IEEE Computer Society Press, Los Alamitos (2005)
4. Kuhn, F., Moscibroda, T., Wattenhofer, R.: Fault-Tolerant Clustering in Ad Hoc and Sensor Networks. In: Proceedings 26th International Conference on Distributed Computing Systems (ICDCS) (2006)
5. Koskinen, H., Karvo, J., Apilo, O.: On improving connectivity of static ad-hoc networks by adding nodes. In: Proceedings of the 4th annual Mediterranean Workshop on Ad Hoc Networks (Med-Hoc-Net), pp. 169–178 (2005)
6. Li, Y.S., Thai, M.T., Wang, F., Yi, C.-W., Wan, P.-J., Du, D.-Z.: On greedy construction of connected dominating sets in wireless networks. Wiley Journal on Wireless Communications and Mobile Computing 5(8), 927–932 (2005)
7. Shang, W.-P., Yao, F., Wan, P.-J., Hu, X.-D.: Algorithms for minimum m-connected k-tuple dominating set problem. Theoretical Computer Science (to submitted)
8. Sinha, P., Sivakumar, R., Bharghavan, V.: Enhancing ad hoc routing with dynamic virtual infrastructures. In: Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1763–1772 (2001)
9. Tarjan, R.: Depth first search and linear graph algorithms. SIAM Journal on Computing 1(2), 146–160 (1972)
10. Wan, P.-J., Alzoubi, K.M., Frieder, O.: Distributed construction of connected dominating set in wireless ad hoc networks. Mobile Networks and Applications 9(2), 141–149 (2004)
11. Wang, F., Thai, T.: On the construction of 2-connected virtual backbone in wireless networks. IEEE Transactions on Wireless Communications (to appear)
12. Wu, W., Du, H., Jia, X., Li, Y., Huang, C.-H.: Minimum connected dominating sets and maximal independent sets in unit disk graphs. Theoretical Computer Science 352(1), 1–7 (2006)