Optim Lett (2011) 5:297–306
DOI 10.1007/s11590-010-0208-8

ORIGINAL PAPER

An exact algorithm for minimum CDS with shortest path constraint in wireless networks

Ling Ding · Xiaofeng Gao · Weili Wu · Wonjun Lee · Xu Zhu · Ding-Zhu Du

Received: 20 May 2010 / Accepted: 24 May 2010 / Published online: 9 June 2010

Abstract In this paper, we study a minimum connected dominating set problem (CDS) in wireless networks, which selects a minimum CDS with property that all intermediate nodes inside every pairwise shortest path should be included. Such a minimum CDS (we name this problem as SPCDS) is an important tache of some other algorithms for constructing a minimum CDS. We prove that finding such a minimum SPCDS can be achieved in polynomial time and design an exact algorithm with time complexity $O(\delta^2 n)$, where δ is the maximum node degree in communication graph.

Keywords CDS · Shortest path · Exact algorithm

L. Ding (✉) · X. Gao · W. Wu · D.-Z. Du
Department of Computer Science, The University of Texas at Dallas, Dallas, USA
e-mail: ling.ding@student.utdallas.edu

X. Gao
e-mail: xiaofenggao@utdallas.edu

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

D.-Z. Du
e-mail: dzdu@utdallas.edu

W. Lee
Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea
e-mail: wlee@korea.ac.kr

X. Zhu
Department of Mathematics, Xi'an Jiaotong University, Xi'an, People's Republic of China
e-mail: zhuxu@mail.xjtu.edu.cn

1 Introduction

In wireless networks, it is very difficult to realize routing on-demand or table-driven routing [1], since the topology is changing from time to time and the energy of each node is very limited. Motivated by the physical backbone in wired networks, a connected dominating set (CDS) is imposed as a virtual backbone [2] on wireless networks, which can make routing in wireless networks more efficient and practical. We can shrink the searching space for routing problem from the whole network to a CDS to reduce routing time and routing table size. Besides routing protocols, CDS can also be used in coverage problem [3], broadcasting [4] and many other network applications. Due to so many benefits, CDS has attracted much attention in recent years.

CDS is a subset of nodes in networks and it divides node set into two parts. Nodes inside CDS form a connected sub-network, which is in charge for routing process. Every node outside CDS should have at least one adjacent node in this CDS. Thus, node outside CDS will always acquire routing path through this neighbor wherever its destination is. The performance of a CDS for coverage, routing, and broadcasting, etc. depends on the size of this CDS. The smaller the size is, the less the routing time will be, and the smaller the routing table size is. Thus much work is devoted to reducing the size of CDS. However, computing a minimum CDS is NP-hard [5], and approximation algorithm is proposed in [5–7].

On one hand, Wu et al. [6] proposed a simple but efficient algorithm for constructing a CDS. Wu's algorithm consists of two steps. Firstly, a CDS was constructed to include all intermediate nodes of all pairwise shortest paths in a network. Next, pruning some nodes in the CDS constructed in the first step to reduce the size of CDS. Wu's algorithm was designed in a distributed way by local information. On the other hand, time complexity, message complexity, and size of CDS were reduced dramatically compared to other algorithms.

However, in [6], they did not check whether we can find a minimum SPCDS in polynomial time or not. In fact, constructing a minimum SPCDS is an important and indispensable step in Wu's algorithm—the first step. In this paper, we will prove that such a minimum SPCDS can be found in polynomial time which means we can find a minimum CDS including all intermediate node of all pairwise shortest paths in a network in polynomial time. Since the problem of minimum SPCDS is solvable in polynomial time, we will introduce two algorithms to construct a minimum SPCDS in polynomial time.

On the other hand, in routing protocols, a shortest path [1] is the first choice for communications between two nodes, because it involves fewest number of nodes in routing path. However, in previous CDS related work, researchers only focused on finding a minimum size CDS. They did not consider the property for routing. Their selected CDS may increase routing cost heavily. For example, in Fig. 1, node A, B, I and G are chosen as a minimum CDS. The original shortest path between D and F is of length 2 ($\{D, E, F\}$) in the network, but the path between D and F through CDS will increase to 5 ($\{D, A, B, I, G, F\}$). Correspondingly, message transmission failure, energy consumption, interference and delay time will increase.

Mohammed et al. [8] pointed out that in wireless networks, the probability of message transmission failure often increases when a message is sent through a longer

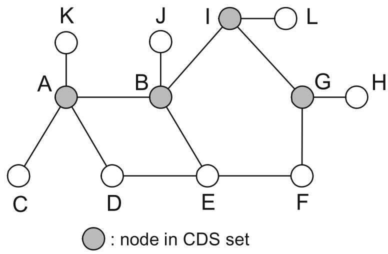

Fig. 1 An example CDS set

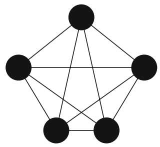

Fig. 2 An example of a minimum SPCDS set in complete graph

path. They mentioned a concept of diameter, which represents the length of the longest shortest paths between a pair of nodes in a given connected network. According to this concept, work has been done to deal with how to select a minimum CDS with smallest diameter of sub-network induced by CDS [9]. Later, Kim et al. [10] proposed the concept of average backbone path length (ABPL), which is the average length of the shortest path between any two nodes in CDS. ABPL is proposed because the authors argue that diameter only considers the worst case of the network, but improving the worst case may not improve the overall performance.

However, such a virtual backbone even considering diameter or ABPL, may increase length of path between some pairs of nodes. Thus unfortunately, packets will be delivered on a longer path through CDS. In addition, from the aspect of fault-tolerance issue, more than one shortest path should be included in a virtual backbone for robustness and efficiency. In our paper, similar to [6], we consider a problem named shortest path connected dominating set (SPCDS), which is a minimum node set including all intermediate nodes in every path between any two nodes in the network. Due to this constraint, the length of path between any two nodes will not be increased even if the path is constructed through SPCDS. According to the definition of SPCDS, in Fig 1, A, B, I, G, F, E, and D will be chosen as a SPCDS, since all of them are the intermediate nodes on the shortest path between other two nodes in the network. The path between D and F through SPCDS will be of length 2 ($\{D, E, F\}$). Since nodes in wireless networks are easy to fail, including all intermediate nodes of all shortest paths in SPCDS is to increase fault tolerance ability of SPCDS. All in all, we sacrifice the size of CDS to improve the network's performance.

Note that, in this paper, we study how to construct a minimum SPCDS in a network which cannot be modeled as a complete graph. However, it does not mean SPCDS has no solution in a complete graph because any single node can be selected as a minimum SPCDS. In Fig. 2, every pair of nodes can communication directly. Thus, any single node is an optimal solution to SPCDS problem.

The rest of the paper will be organized as follows: In Sect. 2, we will introduce the related work on CDS. In Sect. 3, we will introduce the model we use in this paper and define the problem of SPCDS in details. In Sect. 4, we will prove that a minimum SPCDS can be constructed in polynomial time and present our 2-hop neighborhood information based algorithm. Finally, the paper is concluded in Sect. 5.

2 Related work

The research work on selecting a minimum CDS has never been interrupted because of its dramatic contributions to wireless networks. It has been proved that selection of a minimum CDS in a general graph is an NP-hard problem [11] and it is even an NP-hard problem in unit disk graph (UDG) [12]. Thus, many approximation algorithms have been proposed to construct a CDS with a better approximation ratio.

We will introduce constructions of CDS from two aspects—centralized constructions and distributed constructions.

We first introduce some centralized algorithms for selecting minimum CDS. We can category centralized CDS algorithms into two types—one is 1-stage and the other is 2-stage. In 2-stage algorithms, the first step is to select a minimum dominating set (DS) and the second step is to construct a minimum CDS using the technique of Steiner Tree [13]. DS is a subset of nodes in original network, where nodes outside DS have at least one adjacent node inside DS. Different from CDS, subnetwork induced by DS may be disconnected. In contrast, 1-stage algorithms aim to select a CDS directly, skipping the step of finding a DS. In [14], two centralized greedy algorithms were proposed. The first algorithm is 1-stage strategy with approximation ratio of $2H(\delta) + 2$ where δ is the maximum node degree in the network and H is harmonic function. The second strategy proposed in [14] is a 2-stage strategy and yields a approximation ratio of $H(\delta) + 2$. Later, based on the main idea of [14], Ruan et al. [15] made a modification of the selection standard of DS. Therefore, 2-stage is reduced to 1-stage, with approximation ratio of $3 + ln(\delta)$. Recently, Min et al. [16] applied maximum independent set (MIS) to the selection of minimum DS because MIS is also a minimum DS in undirectional graph. Min et al. [16] used an approximation algorithm proposed by [5] for selecting MIS to obtain a minimum DS with size of $3.8|OPT| + 1.2$ and Steiner Tree with minimum number of Steiner nodes (ST-MSN) [17], was used in the second stage. In [16], Min et al. extended the 3-approximation algorithm in Euclidean plane [17] to a unit-disk graph while keeping the approximation ratio the same. This extended algorithm was applied to construct a Steiner Tree in which terminal points are nodes selected from the first stage. As a result, Min achieved an algorithm for selecting a minimum CDS with size of $6.8|OPT|$ at most.

Due to the inefficiency of centralized algorithms in computation, distributed algorithms are much more attractive than centralized ones. Motivated by [18], we can divide unweighted CDS into three categories. The first one is greedy CDS construction. Das et al. [19] implemented the two centralized algorithms in [14] in a distributed way. They approximated a minimal CDS C^* with a performance ratio of $2H(\delta)+1$ in $O((n + |C^*|)\delta)$ time, using $O(n|C^*| + m + n \log n)$ messages, where m is the cardinality of the edge set. The second one is DS based CDS construction. Most algorithms in this type are divided into two phases. The first phase is to construct a DS using the technique of MIS. And add more nodes to make DS be a CDS in the second phase using the technique of Steiner Tree. Butenko et al. [20] proposed a Leader algorithm to achieve an approximation ratio of $8|OPT|+1$ same as that in [21] with time complexity of $O(n)$ and message complexity of $O(n \log n)$. The last type should be pruning based CDS construction. The main idea of this type is that a CDS is constructed firstly with many more redundant nodes. Then prune the redundant nodes from selected CDS to construct a minimum CDS. A typical algorithm of this type is that proposed in [6]. They achieved an approximation of $O(n)$ with time complexity of $O(\delta^3)$.

In addition, CDS has many applications in wireless networks. It can be used in routing [22], broadcasting [4], and topology control [23].

3 Problem statement

In this section, we first introduce the mathematical model used in this paper. Based on the model, we will show what our special SPCDS is.

3.1 System model

We model a wireless network as a connected unit disk graph (UDG) $G = (V, E)$ in which V represents the node set and E represents the edge set. There exists an edge between two nodes if and only if both of them are in the transmission range of each other. In an UDG, every node has the same communication range. We use the concept of hop distance (not Euclidean distance) to evaluate the length of each path in this paper. Given a node subset $D \subseteq V$, D is said to be connected if it induces a connected subgraph $G[D]$ from G.

In G, distance between u and v is the number of hops on the shortest path between them, denoted as $H(u, v)$.

3.2 Problem definition

Let $p(u, v) = \{u, w_1, w_2, \ldots, w_k, v\}$ be one shortest path between u and v in V, and all nodes on $p(u, v)$ except u, v are called intermediate nodes. Every node pair may have more than one shortest paths and these shortest paths compose of a shortest path set $P(u, v)$. For instance, in Fig. 1, the shortest path between B and D can be $p_1(B, D) = \{B, E, D\}$ or $p_2(B, D) = \{B, A, D\}$. Therefore, the shortest path set between node B and C should be $P_{B,D} = \{p_1(B, D), p_2(B, D)\}$.

The SPCDS problem can be formally defined as follows:

Definition 1 (*SPCDS*) The shortest path connected dominating set problem (SPCDS) is to find a minimum size node set $S \subseteq V$ such that

1. $\forall\ u, w \in V$ having $H(u, w) \geq 2, \forall\ p_i(u, w) = \{u, v_1, \ldots, v_k, w\} \in P(u, v)$, all intermediate nodes $v_1, v_2, \ldots, v_k$ should belong to S.

Lemma 1 *If G is not a complete graph and S is a subset of nodes satisfying Definition 1, then S is a CDS. If G is a complete graph, then every single nodes is a minimum SPCDS.*

Proof If G is a complete graph, S will be empty according to Definition 1. Then choose any single node in V and the chosen single node is a minimum SPCDS because, all other node in V are dominated by the chosen one, the single node is connected and the chosen node will not violate the definition of SPCDS since there does not exist any pair of nodes $u, v \in V$ having $H(u, v) \geq 2$. Therefore, any chosen node should be a minimum SPCDS.

If G is not a complete graph, S will not be empty.

First, we show that S is a dominating set. For contradiction, suppose S is not a dominating set. Then there exists a node x not dominated by S. Thus, the shortest path from x to S, $\{x, v_1, \ldots, v_k, y\}$, for some $y \in C$, has $H(x, y) \geq 2$. By Definition 1, all intermediate nodes $v_1, \ldots, v_k$ should belong to S and hence x is dominated by v_1. Contradiction happens, thus S is a dominating set.

Next, we show that S induces a connected subgraph. For contradiction, suppose the subgraph $G[S]$ induced by S is not connected. Then S can be partitioned into two parts $S^{'}$ and $S^{''}$ such that the shortest path from $S^{'}$ to $S^{''}$, $\{x, v_1, \ldots, v_k, y\}$ where $x \in S^{'}$ and $y \in S^{''}$ has $H(x, y) \geq 2$. However, by Definition 1, $v_1, \ldots, v_k$ must belong to S, i.e., they either in $S^{'}$ or in $S^{''}$, which implies that $S^{'}$ and $S^{''}$ have distance one. Contradiction happens. Thus, S induces a connected subgraph. □

4 Theoretical analysis and algorithm

In this section, to solve SPCDS, we define a similar problem named 2-hop shortest path connected dominating set (2PCDS). Next, we will prove the two problems are equivalent to each other. Inspired by 2PCDS, we then propose an optimal solution to SPCDS based on 2-hop neighborhood information.

4.1 Problem reduction

To prove the selection of S is solvable in polynomial time, we first introduce another problem with definition as follows:

Definition 2 (*2PCDS*) The 2-hop shortest path connected dominating set problem (2PCDS) is to find a minimum size node set $S \subseteq V$ such that

1. $\forall\, u, w \in V$ having $H(u, w) = 2, \forall\, p_i(u, w) = \{u, v_1, \ldots, v_k, w\} \in P(u, v)$, all intermediate nodes $v_1, v_2, \ldots, v_k$ should belong to S.

Now, we prove that SPCDS and 2PCDS are equivalent.

Lemma 2 *A dominating set S satisfies Definition 1 if and only if it satisfies Definition 2.*

Proof "⇒": If S meets Definition 1, then intermediate node of any shortest path of length 2 should be included in S. It is trivial that S satisfies Definition 2.

"⇐": Conversely, assume S satisfies Definition 2, we show that S also meets Definition 1. Consider a shortest path $p(u, v) = \{u, w_1, w_2, \ldots, w_k, v\}$, every sub-path of length 2 (such as $\{u, w_1, w_2\}$, $\{w_1, w_2, w_3\}$, and $\{w_{i-1}, w_i, w_{i+1}\}$) among any three consecutive nodes on $p(u, v)$ should be a shortest path between the begin node and the end one. This can be proved by contradiction. Assume there is one subpath $\{w_{i-1}, w_i, w_{i+1}\}$ on $p(u, v)$ which is not the shortest path between w_{i-1} and w_{i+1}, then the shortest path between them must be $\{w_{i-1}, w_{i+1}\}$. Then we can get that $p(u, v)$ is not a shortest path between u, v since we can replace the path $\{w_{i-1}, w_{i+1}\}$ of $\{w_{i-1}, w_i, w_{i+1}\}$ on $p(u, v)$ (contradiction happens). Therefore according to Definition 2, every intermediate node w_i for path $\{w_{i-1}, w_i, w_{i+1}\}$ should be included in S. Then every intermediate node in $p(u, v)$ is included in S, resulting Definition 1.

Next, we show how we can get an optimal solution to SPCDS and 2PCDS.

Lemma 3 *Let S^* be an optimal solution holds Definition 1. Then a node w belongs to S^* if and only if w has two neighbors u and v; and they are not adjacent.*

Proof "$\Leftarrow$": If such two neighbors u and v exist, then w is on the shortest path between u and v. Hence, $w \in S^*$. Such node set S^* meets Definition 2 so that it also meets Definition 1.

"$\Rightarrow$": Conversely, we need to prove if $w \in S^*$, then w must be an intermediate node on a shortest path $p(u, v) = \{u, w, v\}$. This means that w must have two dis-adjacent neighbors u, v. By contradiction, all neighbors for w are adjacent to each other. Then w can be removed from S^* without changing its property, which contradicts to the hypothesis that S^* is optimal.

4.2 Algorithm design

According to Lemma 2, a solution to 2PCDS is also a solution to SPCDS. According to Lemma 3, if we want to decide whether a node v belongs to the solution to SPCDS, we only need to check whether it has two dis-adjacent neighbors. Moreover, to determine this requirement, we only need to check the local neighborhood information for v. Therefore, we design an algorithm using this idea. Firstly, we represent what 2-hop information means and how to maintain it. And then we will introduce a solution to SPCDS, inspired by 2PCDS.

4.2.1 2-Hop neighborhood information maintenance

Each node v sends "Hello" messages to its neighbors. Each "Hello" message is piggybacked with the sender v's information. By collecting "Hello" messages from its neighbors for the first time, v obtains information about its 1-hop neighbors set $N(v)$ (exclude itself). However v has no idea about the relationship among its neighbors. In the following interval, by exchanging 1-hop neighbor information $N(v)$, 2-hop neighbor information $N^2(v)$ is constructed. Specially, $N^2(v) = N(v) \cup \bigcup_{u \in N(v)} N(u)$. According to $N^2(v)$, v can decide whether two nodes u and w in $N(v)$ have a link (u, w). We use Fig. 3 to illustrate our process. In Fig. 3(e), based on 2-hop neighborhood information $N^2(E)$, a subgraph is obtained.

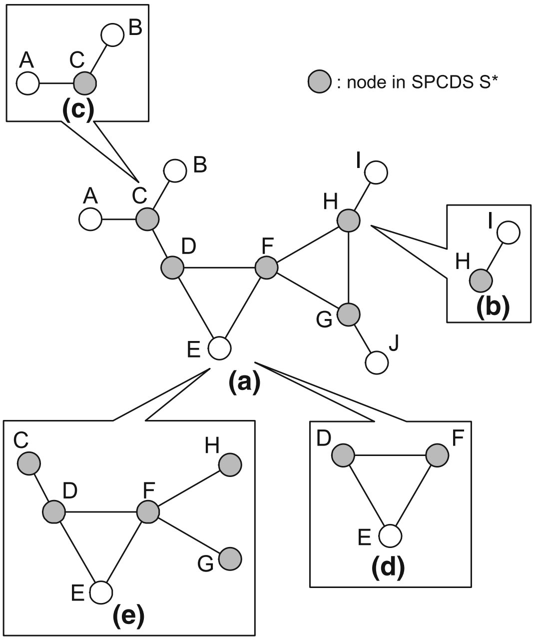

Fig. 3 An example network with 10 nodes. **a** S^*. **b** Node I's view. **c** Node C's view. **d** Node E's view. **e** Subgraph induced from $N^2(E)$

4.2.2 Algorithm

We propose Algorithm 1 based on 2-hop neighborhood information. After collecting information, nodes will be checked one by one to see whether it has a pair of dis-adjacent neighbors. If a node has only one 1-hop neighbor, then the node will not be selected as one member in S^*. In Fig. 3(b), I has only one neighbor H, so it will not be chosen as a node in S^*. In Fig. 3(c), C has two dis-adjacent neighbors A and B, so C should be selected into S^*. In Fig. 3(d), E has two neighbor nodes D and F and, D and F are within each other's transmission range, so E will not be chosen. The detailed algorithm can be shown in Algorithm 1.

Algorithm 1 Selection of S^*

Input : a graph $G = (V, E)$
Output: a subset of V denoted as S^*
Each $v \in V$ sends "Hello" twice to collect $N^2(v)$.
for each $v \in V$ **do**
  if $N(v) \neq \emptyset$ **then**
    for each $u, w \in N(w)$ **do**
      if $w \notin N(u)$ **then** $S^* \leftarrow S^* \bigcup \{v\}$; break;
    end for
  end if
end for

Theorem 1 *The solution of Algorithm 1 is optimal for SPCDS.*

Proof According Lemma 3, by Algorithm 1 Line 4 to Line 6, S^* satisfies Definition 1. In addition, no node in S^* can be deleted. Every node in S^* is selected because it is an intermediate node in one shortest path, so if one node is deleted from S^*, then there exist one shortest path not all intermediate nodes on the shortest path will belong to S^*. As a result, no node can be added or deleted from S^* which means it is optimal. □

Theorem 2 *S^* can be constructed in time $O(\delta^2 n)$, where n is the number of nodes and δ is the maximum node degree of input graph.*

Proof In Algorithm 1, there are two "for" loop. The running time of first "for" (Line 2 to 8) is $O(n)$. And the second running time (Line 4 to 6) is $O(\delta^2)$. In sum, the total running time of the two "for" is $O(\delta^2) \times O(n) = O(\delta^2 n)$. □

5 Conclusion

In this paper, we study SPCDS which is a special case of CDS with shortest path constraint. Due to such constraint, transmission failure, routing delay, and energy cost, etc. will be decreased dramatically because every pairwise path is shortest. Such a CDS is also robust for fault-tolerance. It is well known that finding a minimum CDS is NP-hard, however, we prove that finding a minimum SPCDS is solvable in polynomial time. We also provide an exact algorithm with time complexity $O(\delta^2 n)$, where δ is the maximum node degree of G. In the future work, we may consider reducing the size of SPCDS to make our virtual backbone more efficient.

Acknowledgments This research was supported by National Science Foundation of USA under Grant CNS0831579 and CCF0728851. This research was also jointly supported by MEST, Korea under WCU (R33-2008-000-10044-0), by KRF Grant funded by (KRF-2008- 314-D00354), and by MKE, Korea under ITRC IITA-2009-(C1090-0902-0046) and IITA-2009-(C1090-0902-0007).

References

1. Tsai, Y.-P., Hsu, T.-L., Liu, R.-S., Ho, Y.-K.: A backbone routing protocol based on the connected dominating set in ad hoc networks. World Cong. Comput. Sci. Inf. Eng. **1**, 14–18 (2009)
2. Cheng, X., Ding, M., Du, D.H., Jia, X.: Virtual backbone construction in multihop ad hoc wireless networks: Research articles. Wireless Commun. Mobile Comput. **6**(2), 183–190 (2006)
3. Yiwei, W., Chunyu, A., Shan, G., Yingshu, L.: p-Percent coverage in wireless sensor networks. In: WASA '08: Proceedings of the third international conference on wireless algorithms, systems, and applications, pp. 200–211. Springer, Berlin (2008)
4. Iqbal, A., Ahmed, N., Mostofa Akbar, M.: Directional antenna based connected dominating set construction for energy efficient broadcasting in wireless ad hoc network. Int. Conf. Comput. Electr. Eng. **0**, 839–843 (2008)
5. Wu, W., Du, H., Jia, X., Li, Y., Huang, S.C.-H.: Minimum connected dominating sets and maximal independent sets in unit disk graphs. Theor. Comput. Sci. **352**(1), 1–7 (2006)
6. Wu, J., Dai, F., Gao, M., Stojmenovic, I.: On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. IEEE/KICS J. Commun. Netw. **4**, 59–70 (2002)
7. Cheng, X., Huang, X., Li, D., Wu, W., Du, D.-Z.: A polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks. Networks **42**, 202–208 (2003)
8. Mohammed, K., Gewali, L., Muthukumar, V.: Generating quality dominating sets for sensor network. In: ICCIMA '05: Proceedings of the sixth international conference on computational intelligence and multimedia applications, pp. 204–211. IEEE Computer Society, Washington, DC (2005)

9. Zhang, N., Shin, I., Zou, F., Wu, W., Thai, M.T.: Trade-off scheme for fault tolerant connected dominating sets on size and diameter. In: FOWANC '08: Proceeding of the 1st ACM international workshop on foundations of wireless ad hoc and sensor networking and computing, pp. 1–8. ACM, New York (2008)
10. Kim, D., Wu, Y., Li, Y., Zou, F., Du, D.-Z.: Constructing minimum connected dominating sets with bounded diameters in wireless networks. IEEE Trans. Parallel Distrib. Syst. **20**(2), 147–157 (2009)
11. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of np-Completeness. Series of Books in the Mathematical Sciences. W. H. Freeman, San Francisco (1979)
12. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discret. Math. **86**(1–3), 165–177 (1990)
13. Du, D.-Z., Lu, B., Ngo, H., Pardalos, P.M.: Steiner tree problems. In: Floudas, C., Pardalos, P. (eds.) Encyclopedia of Optimization, vol. 5, pp. 227–290. Kluwer (2001)
14. Guha, S., Khuller, S.: Approximation algorithms for connected dominating sets. Algorithmica **20**(4), 374–387 (1996)
15. Ruan, L., Du, H., Jia, X., Wu, W., Li, Y., Ko, K.-I.: A greedy approximation for minimum connected dominating sets. Theor. Comput. Sci. **329**(1–3), 325–330 (2004)
16. Min, M., Du, H., Jia, X., Huang, C.X., Huang, S.C.-H., Wu, W.: Improving construction for connected dominating set with steiner tree in wireless sensor networks. J. Glob. Optim. **35**(1), 111–119 (2006)
17. Chen, D., Du, D.-Z., Hu, X.-D., Lin, G.-H., Wang, L., Xue, G.: Approximations for steiner trees with minimum number of steiner points. J. Glob. Optim. **18**(1–3), 17–33 (2000)
18. Du, D.-Z., Pardalos, P.: Handbook of Combinatorial Optimization. Kluwer, Dordrecht (2004)
19. Das, B., Bharghavan, V.: Routing in ad-hoc networks using minimum connected dominating sets. In: Proceedings of IEEE ICC (1997)
20. Butenko, S., Cheng, X.Z., Du, D.-Z., Pardalos, P.M.: On the construction of virtual backbone for ad hoc wireless network. In: Proceedings of conference on cooperative control and optimization (2001)
21. Alzoubi, K., Wan, P.-J., Frieder, O.: New distributed algorithm for connected dominating set in wireless ad hoc networks. In: Proceedings of IEEE HICSS (2002)
22. Di, W., Yan, Q., Ning, T.: Connected dominating set based hybrid routing algorithm in ad hoc networks with obstacles. In: Proceedings of IEEE ICC (2006)
23. Deb, B., Bhatnagar, S., Nath, B.: Multi-resolution state retrieval in sensor networks. In: Proceedings of IEEE international workshop on sensor network protocols and applications (2003)