

## AN EFFICIENT CONNECTED DOMINATING SET ALGORITHM IN WSNs BASED ON THE INDUCED TREE OF THE CROSSED CUBE

JING ZHANG <sup>a,b</sup>, LI XU <sup>a,\*</sup>, SHU-MING ZHOU <sup>a</sup>, WEI WU <sup>a</sup>, XIUCAI YE <sup>c</sup>

<sup>a</sup>College of Mathematics and Computer Science, Fujian Normal University  
No. 8, Shangsang Road, Cangshan District, Fuzhou, Fujian, 350007 China  
e-mail: xuli@fjnu.edu.cn

<sup>b</sup>School of Information Science and Engineering, Fujian University of Technology  
3th Xueyuan Road, University Town, Minhou, Fuzhou, Fujian, 350108 China

<sup>c</sup>Department of Computer Science, University of Tsukuba  
1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8577 Japan

The connected dominating set (CDS) has become a well-known approach for constructing a virtual backbone in wireless sensor networks. Then traffic can be forwarded by the virtual backbone and other nodes turn off their radios to save energy. Furthermore, a smaller CDS incurs fewer interference problems. However, constructing a minimum CDS is an NP-hard problem, and thus most researchers concentrate on how to derive approximate algorithms. In this paper, a novel algorithm based on the induced tree of the crossed cube (ITCC) is presented. The ITCC is to find a maximal independent set (MIS), which is based on building an induced tree of the crossed cube network, and then to connect the MIS nodes to form a CDS. The priority of an induced tree is determined according to a new parameter, the degree of the node in the square of a graph. This paper presents the proof that the ITCC generates a CDS with a lower approximation ratio. Furthermore, it is proved that the cardinality of the induced trees is a Fibonacci sequence, and an upper bound to the number of the dominating set is established. The simulations show that the algorithm provides the smallest CDS size compared with some other traditional algorithms.

**Keywords:** wireless sensor networks, connected dominating set, induced tree, approximation algorithm, crossed cube.

### 1. Introduction

Networks of a number of wireless sensor nodes, named wireless sensor networks (WSNs), may be deployed to sense, gather and process information in a region of interest for a variety of purposes, such as many civilian application areas, including traffic control, healthcare applications, home automation, long-term and low-cost geographical monitoring, and so forth (Zhao *et al.*, 2012; Liao and Li, 2013; Goli, 2012). In WSNs, there is no fixed infrastructure, one sensor can communicate with other sensors within the transmission range. In many applications, sensors communicate via a shared medium, either through a single hop or multi-hops. The sole energy source of the nodes in WSNs is the battery. Recharging the battery is impossible since the sensor nodes usually

locate in an unpredictable area. A prolonged network lifetime is a critical issue in energy efficiency algorithms for WSNs (Tang *et al.*, 2012; Kim *et al.*, 2009; Zhu *et al.*, 2011; Padmavathy, 2010; Zam and Movahedinia, 2013).

The self-organization of WSNs includes no fixed infrastructure, while frequent and hard-to-predict topology changes are the most important issues that must be taken into consideration in WSNs, and topology control is one of the approaches to prolong the network lifetime, reduce interference and packet retransmission (Zhao *et al.*, 2012). Topology control is mainly of two types (Tang *et al.*, 2012): power control and hierarchical topology control. The former is to adjust the transmission range of sensors to construct a network with better properties. Although there is no physical backbone infrastructure, the latter aims to form a virtual backbone by constructing a connected dominating set (CDS). Since

\*Corresponding author

traffic is only forwarded by a virtual backbone, routing is easier and can adapt quickly to topology changes, so it can improve the routing performance and save energy of the networks. For example, broadcasting in a WSN could be simplified by letting each node of a CDS transmit once. The nodes within the CDS are called dominators, the nodes that are adjacent to a dominator are called dominatees. WSNs use a CDS to serve as a virtual backbone for energy efficiency.

The concept of a CDS plays a crucial role in the management and maintenance of WSNs, while the size of the CDS is the primary concern to measure the quality of a virtual backbone (Kim *et al.*, 2009). The interference problem will occur because the nodes in WSNs share their communication channel; a smaller virtual backbone suffers less from this problem. Meanwhile, if the CDS is getting smaller, it becomes more efficient when performing in routing. It also makes the maintenance of the virtual backbone easier. So, reducing the size of a CDS is our purpose in this paper. The crossed cube (Cheng *et al.*, 2013) is an important class of variants of hypercubes as rule networks, which is proven to be superior to the hypercube counterpart, since it can reduce the diameter of the network. However, no result has been reported on constructing the CDS by using a crossed cube.

The contribution of this paper is the following:

1. One induced tree of the crossed cube based on an equivalent definition of the crossed cube network is defined, and the induced tree is constructed by induction on the dimension of the crossed cube network.
2. A novel distributed CDS construction algorithm is presented named CDS-ITCC- $G^*$ , which is a CDS construction algorithm based on the induced tree of crossed cube in the square of a graph  $G$ . The algorithm can be easily implemented in practice.
3. The theoretical analysis has been established and clearly shows that CDS-ITCC- $G^*$  can generate a maximal independent set with the approximation ratio of  $3.2833\gamma + 4.5590$ , where  $\gamma$  is the size of any optimal CDS. It is proved that the cardinality of induced trees is a Fibonacci sequence, and an upper bound of the number of the DS is derived.

The rest of this paper is organized as follows. Section 2 briefly summaries related works on the connected dominating set. Section 3 presents the problem statement of our work. Section 4 is devoted to the newly presented CDS CDS-ITCC- $G^*$  algorithm, which is based on the induced tree of the crossed cube. The theoretical analysis and the simulation results are shown in Sections 5 and 6, respectively. Finally, Section 7 concludes the paper.

## 2. Related works

It has been proven that the problem of finding a minimum connected dominating set (MCDS) is NP-hard, but it has a polynomial-time approximation scheme (PTAS) in unit disk graphs (Cheng *et al.*, 2003). That is to say, there is a polynomial-time algorithm producing an approximate solution with the size within a factor of  $1 + \varepsilon$  from an optimal one for  $\varepsilon > 0$ . Many algorithms have been proposed for the purpose of fast approximations with a small performance ratio for constructing an MCDS.

The construction of an MCDS in WSNs has been studied extensively. Some approximation MCDS construction algorithms are introduced in this section, most of them consisting of two stages. The first phase is to construct a dominating set, the second one is to select connectors, in order to link the dominators, and then let the virtual backbone connect (Du *et al.*, 2011; Li *et al.*, 2011). In order to estimate the size of the MCDS, to establish an upper bound for the MIS is critical. Wan *et al.* (2008) presented an example to show that  $3\gamma + 2$  is reachable, while Wu *et al.* (2010) gave a conjecture as the following open problem.

**Open problem.** (*Conjecture*) In a unit disk graph, any maximal independent set has size  $\alpha \leq 3\gamma + 2$ , where  $\alpha$  is the size of the maximal independent set,  $\gamma$  is the size of any optimal CDS.

Each maximal independent set has size at most  $4\gamma + 1$ , which was first shown by Wan *et al.* (2002). Later research on the open problem is still quite active: some efforts (Wu *et al.*, 2006; 2010; Funke *et al.*, 2006; Lin, 2006; Xu and Lin, 2007; Wan *et al.*, 2008; Gao *et al.*, 2009; Li *et al.*, 2011; Han, 2009; Bahaa-Eldin *et al.*, 2012) have been made for improving this bound. Wu *et al.* (2006) proposed an algorithm with an approximation factor of  $3.8\gamma + 1.2$ . Wan *et al.* (2008) proved that the approximation ratio of their algorithm is  $\alpha \leq 3\frac{2}{3}\gamma + 1$ , if  $G$  has at least two nodes. Gao *et al.* (2009) improved the upper bound  $3.478\gamma + 4.874$  with the help of the Voronoi diagram and the Euler formula—the area of boundary and non-boundary  $i$ -polygons  $s'_i$  and  $s_i$  was computed. It got the following equations:  $s_3 = 1.299, s_4 = 1, s_5 =$

Table 1. Upper bound for the maximal independent set in a unit disk graph.

Approx. algorithm	Upper bound for the MIS
Wan <i>et al.</i> (2002)	$4\gamma + 1$
Wu <i>et al.</i> (2006)	$3.8\gamma + 1.2$
Funke <i>et al.</i> (2006)	$3.748\gamma + 9$
Wan <i>et al.</i> (2008)	$3.6667\gamma + 1.3333$
Gao <i>et al.</i> (2009)	$3.478\gamma + 4.874$
Li <i>et al.</i> (2011)	$3.4306\gamma + 4.8185$
Wu <i>et al.</i> (2010)	$3\gamma + 2$ (conjecture)
CDS-ITCC- $G^*$	$3.2833\gamma + 4.5590$

0.9082,  $s_6 = s_7 = \dots = 0.8661$ ;  $s'_3 = 1.1781$ ,  $s'_4 = 0.9717$ ,  $s'_5 = 0.8968$ ,  $s'_6 = 0.8546$ ,  $s'_7 = s'_8 = \dots = 0.8525$ . The performance of these approximations highly depends on the relationship between the size of an MIS ( $\alpha$ ) and the size of minimum CDS ( $\gamma$ ) in graph  $G$ . Here  $\alpha/\gamma$  is called the theoretical bound to approximate the CDS.

A theorem that the rough bound for the MIS and the MCDS is  $S_{\text{MIS}} \leq S_{\text{MCDS}} \leq 2.9425\gamma + 4.1251$  has been given. They obtained an upper bound  $3.478\gamma + 4.874$  by the Euler formula. Li *et al.* (2011) obtained a tighter relation between the independence number and the connected domination number, and employed quite a complicated geometry argument to push this bound further to  $3.4306\gamma + 4.8185$ . To our knowledge, no result has been reported on constructing the CDS by using rule networks. As a rule network, the crossed cube has good properties. In this paper, based on the induced tree of the crossed cube in the square of a graph, a novel distributed CDS construction algorithm named CDS-ITCC- $G^*$  is presented, which can generate a maximal independent set with the approximation ratio of  $3.2833\gamma + 4.5590$ . The evolution of the improvement is shown in Table 1.

There is also some research on the second stage (Xu and Lin, 2007; Kim *et al.*, 2009; Wan *et al.*, 2008; Han, 2009; Bahaa-Eldin *et al.*, 2012; Wu *et al.*, 2010; Zou *et al.*, 2011; Misra and Mandal, 2010; Li *et al.*, 2011; Tang *et al.*, 2012). Constructing a minimum spanning tree is one majority way to connect a dominating set into a connected virtual backbone. Han (2009) proposed the Zone algorithm, with the highest priority in its neighborhood, where the node is selected as the dominator. The priority of nodes can be determined according to the lowest ID node and the highest degree node. In the second phase, between two dominators, the highest-priority node is considered as a connector node in the same zone. Kim *et al.* (2009) proposed the connected dominated sets-bounded diameters-distributed (CDS-BD-D) clustering algorithm, which is a distributed clustering algorithm that comprises two phases. The first one applies the distributed breadth first search (BFS) algorithm, and the second one selects dominators and connectors.

The CDS-BD-D algorithm is interested in using the average backbone path length (ABPL) to evaluate the CDS, which considers energy to extend the network lifetime. The ABPL of a CDS is the sum of the hop distance between any pair of CDSs divided by the number of all the possible pair of nodes. Tang *et al.* (2012) proposed an efficient approximation MCDS construction algorithm, E-MCDS (energy efficient MCDS construction algorithm), which explicitly takes energy consumption into account. The E-MCDS contains the CDS construction stage and the pruning stage. In this paper, the performance of our algorithm is compared

with the CDS-BD-D algorithm (Kim *et al.*, 2009), Zone-Min-ID, Zone-Max-Degree algorithms (Han, 2009) and the E-MCDS (Tang *et al.*, 2012).

The objectives of the algorithms described above include minimizing the size of the CDS, but the objectives of CDS construction are not or not only minimizing the size of the CDS in some research. For example, there exist works (Wang *et al.*, 2009; Thai *et al.*, 2007; Dai and Wu, 2005; Wu *et al.*, 2007; Wu and Li, 2008; Li *et al.*, 2012b; Kim *et al.*, 2010) that focus on constructing  $k$ -connected  $m$ -dominating sets for fault tolerance. Algorithms of Li *et al.* (2012a), Liu *et al.* (2013), He *et al.* (2013), Ding *et al.* (2011) or Wu and Li (1999) focus on reducing routing costs, rather than minimizing the total CDS size. These algorithms have no fixed approximation ratios. Some works consider constructing CDS for heterogeneous networks; for example, the algorithm proposed by Ding *et al.* (2012) takes into the consideration the issue of constructing an energy-efficient virtual network using directional antennas.

### 3. Problem statement

**3.1. Definitions and notation.** In this paper, a communication model will be introduced by using graph theory. Sensor nodes are randomly distributed in the network field and have the same transmission range. The link between any pair of nodes is bidirectional. One network is modeled as a connected bidirectional graph  $G = (V, E)$ , where  $V$  and  $E$  represent the node set and the link set in  $G$ , respectively. For any  $u, v \in V$  there exists an edge  $(u, v)$  in  $G$  if and only if  $u$  is in  $v$ 's transmission range in the network,  $v$  is also in  $u$ 's transmission range, and there is no obstacle preventing radio wave transmission between  $u$  and  $v$ .

**Definition 1. (Node neighbourhoods)** (Bahaa-Eldin *et al.*, 2012) Consider a node  $u$ . The set of nodes covered by  $u$  is represented by  $N(u)$ ,  $N(u) = \{v \mid (v, u) \in E\}$  is called the open neighbor set of  $u$ ,  $N[u] = N(u) \cup \{u\}$  is called the closed neighbor set of  $u$ .

Nodes using an exchange of hello messages can find its distance-1 neighbour nodes and ascertain their degree.  $N_2(u)$  denotes the set of nodes which are at most at distance-2 from  $u$ . The distance-2 neighbour of  $u$  is represented as  $N_2(u) - N(u)$ , which is composed of node  $u$ 's local graph denoted as  $LG(u)$ .

**Definition 2. (Maximal independent set)** (Kim *et al.*, 2010) A maximal independent set of a graph  $G = (V, E)$  is a subset  $V' \subseteq V(G)$  such that every pair of vertices in  $V'$  is not adjacent, and no independent vertex can be added into  $V'$ .

**Definition 3. (Dominating set)** (Kim *et al.*, 2010) A dominating set of a graph  $G = (V, E)$  is a set of nodes

$V' \subseteq V(G)$  such that, for every  $(u, v) \in E(G)$ ,  $u \in V'$  or  $v \in V'$ .

**Definition 4.** (Connected dominating set) (Kim et al., 2010) A connected dominating set of a graph  $G = (V, E)$  is a DS of  $G$  such that the subgraph of  $G$  induced by the nodes in this set is connected.

In many cases, an MIS construction algorithm is used to find a DS. The nodes in the CDS are called dominators and dominatees otherwise. The size of the CDS is equal to the number of the dominators. Our algorithm is to find an MIS which is based on building an induced tree of the crossed cube network, and then connect the MIS nodes to form a CDS.

**3.2. Induced trees of the crossed cube network.**

The ITCC algorithm is a distributed CDS constructed algorithm; the construction of CDS is based on a square of a local graph, which is defined as following.

**Definition 5.** (Square of a graph  $G^2$ ) (Bahaa-Eldin et al., 2012) The square of a graph  $G$ , denoted by  $G^2$ , is a graph on the same vertex set such that two vertices are adjacent in  $G^2$  iff their distance in  $G$  is at most two hops.

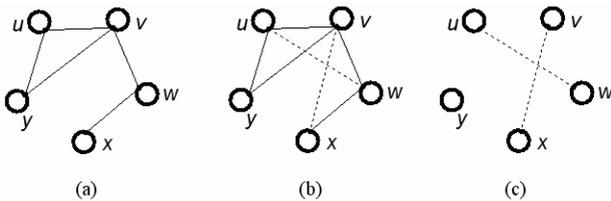


Fig. 1. Graphs  $G$  (a), the square of graph  $G^2$  (b), the star of graph  $G^*$  (c).

As shown in Fig. 1(a),  $N(u) = \{v, y\}$ ,  $N(v) = \{u, y, w\}$ ,  $N(w) = \{v, x\}$ ,  $N(x) = \{w\}$ . Since  $u, w \in N(v)$ , but  $w \notin N(u)$ , add the link  $(u, w)$  when  $G^2$  is constructed; in a similar way,  $(v, x)$  is the added link, which is shown in Fig. 1(b). Set  $G^* = G^2 \setminus E(G)$ , as depicted in Fig. 1(c). Our algorithm is a distributed local algorithm, so all the following  $G^*$ 's are constructed on some local graphs  $LG(u)$ .

**Definition 6.** (Crossed cube) (Cheng et al., 2013) The crossed cube  $CQ_1$  is a complete graph with two vertices labeled by 0 and 1, respectively. For  $n \geq 2$ , an  $n$ -dimensional crossed cube  $CQ_n$  consists of two  $(n - 1)$ -dimensional sub-crossed cubes,  $CQ_{n-1}^0$  and  $CQ_{n-1}^1$ , and a perfect matching between the vertices of  $CQ_{n-1}^0$  and  $CQ_{n-1}^1$ , according to the following rule.

Let  $V(CQ_{n-1}^0) = \{0u_{n-2}u_{n-3} \dots u_0 \mid u_i = 0 \text{ or } 1\}$  and  $V(CQ_{n-1}^1) = \{1v_{n-2}v_{n-3} \dots v_0 \mid v_i = 0 \text{ or } 1\}$ .

The vertex  $u = 0u_{n-2}u_{n-3} \dots u_0 \in V(CQ_{n-1}^0)$  and the vertex  $v = 1v_{n-2}v_{n-3} \dots v_0 \in V(CQ_{n-1}^1)$  are adjacent in  $CQ_n$  if and only if

- (1)  $u_{n-2} = v_{n-2}$  if  $n$  is even, and
- (2)  $u_{2i+1}u_{2i} \sim v_{2i+1}v_{2i}$ , for  $0 \leq i \leq \lfloor (n - 1)/2 \rfloor$ .

Two binary strings  $u = u_1u_0$  and  $v = v_1v_0$  are pair-related, which is denoted by  $u \sim v$ , if

$$(u, v) \in \{(00, 00), (01, 11), (11, 01), (10, 10)\}.$$

An edge  $(u, v) \in E(CQ_n)$  is labelled by  $j$  if  $u_j \neq v_j$  and  $u_i = v_i$  for  $j + 1 \leq i \leq n - 1$ . For  $n \geq 2$ ,  $\text{suff}(u) = u_1u_0$  denotes the rightmost two bits of  $u$ .

For example,  $CQ_1$ ,  $CQ_2$ ,  $CQ_3$  and  $CQ_4$  are shown in Figs. 2(a1), (b1), (c1) and (d1), respectively.

**Definition 7.** (Vertex extension) (Wang et al., 2011) Let  $u$  be a binary string. A quadrilateral consisting of  $u00$ ,  $u01$ ,  $u11$ ,  $u10$  is called a basic quadrilateral of  $u$ , denoted by  $C_u$ . The edges  $\{(u00, u01), (u01, u11), (u11, u10), (u10, u00)\}$  in  $C_u$  are called basic edges.

For all  $n \geq 3$ ,  $CQ_n$  is a crossed extension of all the edges in  $CQ_{n-2}$ .

All the trees are bipartite, the bipartite partition of the tree  $T$  is denoted as  $\{V_1(T), V_2(T)\}$ . That is  $V_1(T), V_2(T) \subseteq V(T)$ , which satisfies  $V_1(T) \cap V_2(T) = \phi$  and  $V_1(T) \cup V_2(T) = V(T)$ . For every  $(u, v) \in E(T)$ , there exist  $u \in V_1(T), v \in V_2(T)$ .  $|V_1(T)| \leq |V_2(T)|$  is set without loss of generality.

**Definition 8.** (Induced trees of the crossed cube) The induced tree of  $CQ_1$  is a tree with two vertices labelled by 0 and 1, respectively,

$$V_1(T) = \{1\}, V_2(T) = \{0\}.$$

The induced tree of  $CQ_2$  is a tree with three vertices labelled by 00, 01 and 10, respectively,

$$V_1(T) = \{00\}, V_2(T) = \{01, 10\}.$$

Let  $T_n$  be an induced tree of  $CQ_n (n \geq 2)$ . Set

$$\begin{aligned} V_1(T_{n+2}) &= \{u01 \mid u \in V_1(T_n)\} \\ &\cup \{u00 \mid u \in V_2(T_n)\}, \\ V_2(T_{n+2}) &= \{u10, u01 \mid u \in V_2(T_n)\} \\ &\cup \{u00 \mid u \in V_1(T_n)\}. \end{aligned}$$

Then  $T_{n+2}$  is the induced tree of  $CQ_{n+2}$  with the set  $V_1(T_{n+2}) \cup V_2(T_{n+2})$ .

The induced trees  $T_n (n = 1, 2, 3, 4)$  are shown in Figs. 2(a2), (b2), (c2) and (d2), respectively, where

$$\begin{aligned} V(T_1) &= \{0, 1\}, V(T_2) = \{00, 01, 10\}, \\ V(T_3) &= \{000, 001, 010, 100, 101\}, \end{aligned}$$

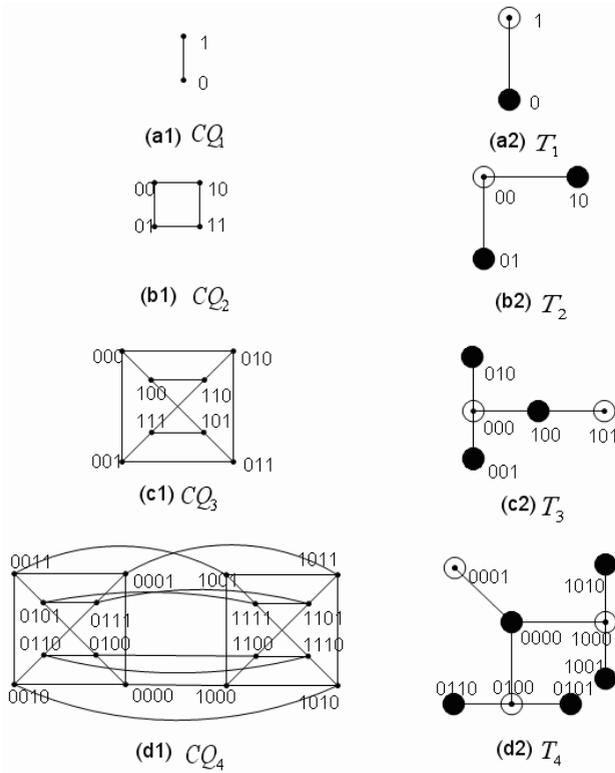


Fig. 2. Graphs  $CQ_1$  (a1),  $CQ_2$  (b1),  $CQ_3$  (c1),  $CQ_4$  (d1), trees  $T_1$  (a2),  $T_2$  (b2),  $T_3$  (c2),  $T_4$  (d2).

$$V(T_4) = \{0000, 0001, 0100, 0101, 0110, 1000, 1001, 1010\}.$$

Assume the set of hollow circles is  $V_1(T_n)$ , and the set of solid circles is  $V_2(T_n)$ .

#### 4. CDS-ITCC- $G^*$ algorithm

In order to understand CDS-ITCC- $G^*$  better, the overview of the algorithm is given first. Then we describe each part of the algorithm in detail. There are mainly three stages in CDS-ITCC- $G^*$ : the first one is to construct  $G^*$ , the second is to find the MIS based on the ITCC, which is constructed in the local graph  $G^*$ , and then form a CDS by connecting the nodes in MIS. The third stage is to prune the set with the elimination rule, which aims to minimize the size of the CDS.

**4.1.  $G^*$  construction stage.** At the initial stage, all nodes have to be given an initial status and then exchange neighbour information among themselves to get ready for CDS construction. All nodes must construct the local graphs  $G$ ,  $G^2$  and  $G^*$  with themselves as the centre. Each node  $u_i$  holds message (id,  $W(u_i)$ ), where  $W(u_i) = (f(u_i), d_{G^*}(u_i), d_G(u_i), re(u_i))$  is the weight

of the corresponding node. For each node  $u_i$ ,  $f(u_i)$  is the identification of the node status, the numerical value can be obtained by the formula:

$$f(u_i) = \begin{cases} 0, & \text{if } u_i \text{ is a normal node,} \\ 1, & \text{if } u_i \text{ is in the DS,} \\ 2, & \text{if } u_i \text{ is a connected node,} \\ 3, & \text{if } u_i \text{ has been eliminated.} \end{cases} \quad (1)$$

Here  $d_{G^*}(u_i)$  is the degree of the node  $u_i$  in graph  $G^*$ ,  $d_G(u_i)$  is its degree in graph  $G$ , and the value of  $re(u_i)$  represents residual energy. The node's weight is the priority for these elements. For example, for some nodes with the same value of  $f$ , the higher the degree of  $d_{G^*}$ , the greater the weight, so the probability of the node to be selected into the CDS is greater. A detailed description will be given in the following algorithm.

Following the stages described in Sections 4.2 and 4.3 is the main operation for generating a near-optimal MCDS, which is composed of two steps: CDS-ITCC- $G^*$  construction and the elimination stage.

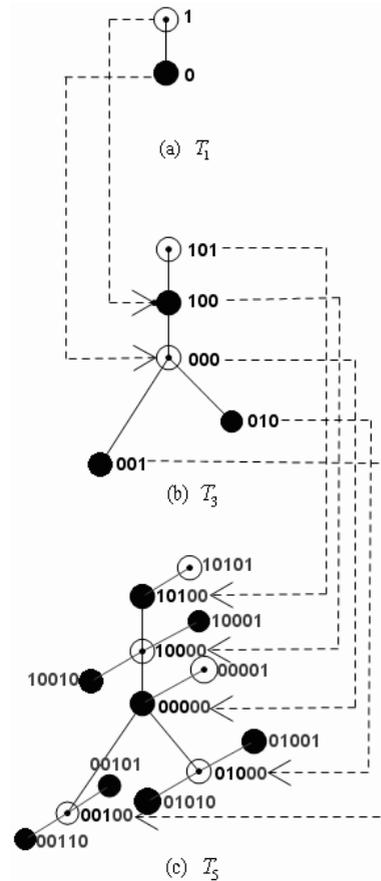


Fig. 3. Construction of the induced tree of the crossed cube  $T_1$  (a),  $T_3$  (b),  $T_5$  (c).

**4.2. CDS-ITCC- $G^*$  construction stage.** The CDS-ITCC- $G^*$  construction stage is divided into two sub-stages, which are a DS-ITCC- $G^*$  construction stage and a connecting stage. In the first sub-stage, an initial node is selected as the root, and then, to construct an induced tree of the crossed cube, the second sub-stage aims to connect the DS and form a CDS. Figure 4 shows the overall mechanism of the CDS-ITCC- $G^*$  algorithm.

The  $(n + 2)$ -dimensional induced tree  $T_{n+2}$  is uniquely determined by the  $(n)$ -dimensional induced tree  $T_n$ . The 1- and 2-dimensional induced trees  $T_1, T_2$  are determined in Definition 8. Then the induced tree of any dimensions crossed cube can be constructed recursively as

$$T_1 \rightarrow T_3 \rightarrow T_5 \dots$$

and

$$T_2 \rightarrow T_4 \rightarrow T_6 \dots$$

Figure 3 shows an example of the recursive procedure of induced trees.  $T_1$  is shown in Fig. 3(a), where  $V_1(T_1) = \{1\}$  and  $V_2(T_1) = \{0\}$  are the bipartite partitions of the tree  $T_1$ . Figure 3(b) is the induced tree  $T_3$ . Since

$$\begin{aligned} V_1(T_{n+2}) &= \{u01 \mid u \in V_1(T_n)\} \\ &\quad \cup \{u00 \mid u \in V_2(T_n)\}, \\ V_2(T_{n+2}) &= \{u10, u01 \mid u \in V_2(T_n)\} \\ &\quad \cup \{u00 \mid u \in V_1(T_n)\}, \end{aligned}$$

the set of hollow circles is  $V_1(T_3) = \{000, 101\}$ , and the set of solid circles is  $V_2(T_3) = \{001, 010, 100\}$ . Similarly, the set of hollow circles shown in Fig. 3(c) is  $V_1(T_5)$ , and the set of solid circles is  $V_2(T_5)$ , which is extended from  $T_3$ . Algorithm 1 is a CDS algorithm based on induced trees of the crossed cube network.

Initialization for the parameters should be conducted before the algorithm starts (Line 1 in Algorithm 1), then, one node is selected to join into the graph of induced trees of the crossed cube  $G_T$  (Line 2 in Algorithm 1). Assume that the node  $u_0$  has the maximum weight, labelled by 0. The function  $\text{dom}(V(T_n))$  is called at the same time. It aims to find the dominatees for each dominator in  $V(T_n)$ , and the set of  $N_G(V(T_n))$  is added into the set of the dominatees  $V_d$ . Since a flag named  $\text{flag}$  is used to describe whether the algorithm is completed, it is also implemented by the function  $\text{dom}()$ , which is to judge whether all the nodes in the network have decided their own states, as a dominator or a dominatee. When  $V_d \cup S$  is equal to  $V$ , this means the algorithm is completed, and set  $\text{flag} \leftarrow 1$ .

If  $\text{flag}! = 1$ , the function  $\text{one-tree}()$  is called to construct the 1-dimension induced tree of the crossed cube  $T_1$ . One node  $u_1$ , which has the maximum weight, is selected from  $N_{G^*}(u_0)$ , labeled by 1, and set  $f(u_1) \leftarrow 1$ . This new node  $u_1$  and the related edge are added into induced tree  $V(T_1)$  at the same time, then, the new

**Algorithm 1.** CDS algorithm based on induced trees of the crossed cube network (CDS-ITCC- $G^*$ ).

---

**Require:** A simple connected graph  $G(V, E)$

- 1:  $V(T_n) = \phi; E(T_n) = \phi; CDS = \phi$ , the set of the dominatees  $V_d = \phi$ ; each node  $u_i$  hold  $(\text{id}, W(u_i))$ , where  $f(u_i) = 0; \text{flag} = 0; n = 1$ .
- 2: select a node  $u_0$ , labeled by 0 with  $f(u_0) = 1, V(T_n) = u_0$ , call function  $\text{dom}(V(T_n))$ ;
- 3: if ( $\text{flag}! = 1$ )
- 4: call function  $\text{one-tree}(V(T_n))$ ;
- 5: endif;
- 6: if ( $\text{flag}! = 1$ )
- 7:  $n = 2$ ;
- 8: call function  $\text{two-tree}(V(T_n))$ ;
- 9: endif;
- 10: while ( $\text{flag}! = 1$ ) do
- 11:  $n++$ ;
- 12: call function  $\text{induced-tree}(V(T_n))$ ;
- 13: end while;

---

induced tree is constructed as  $V(T_n) \leftarrow \{u_0, u_1\}$  and  $E(T_n) \leftarrow E(T_n) \cup (u_0, u_1)$ . According to Definition 2, the dimension of the induced tree can be extended, the nodes must be divided into bipartite partitions. Let  $V_1(T_n) \leftarrow \{u_1\}$  and  $V_2(T_n) \leftarrow \{u_0\}$ .

When at least two nodes are selected into the induced tree as a DS, the function  $\text{connect}(G_T)$  is called, which aims to select the connector nodes to make the DS connected. This function is implemented as follows. For each edge  $(u, v) \in E(G^*)$ , according to the definition of graph  $G^*$ , there exist some nodes in  $N_G(u) \cap N_G(v)$  if  $f(u) == f(v) == 1$ . The one that has the maximum weight, named  $w$ , can be selected as the connector node; thus, node  $w$  joins the set of CDS referred to as  $CDS \leftarrow V(T_n) \cup w$  and set  $f(w) \leftarrow 2$  at the same time; set  $C_1 \leftarrow CDS$ . Later, the function  $\text{dom}(CDS)$  also must be exacted to judge whether the algorithm is completed.

If it is still  $\text{flag}! = 1$ , the function  $\text{two-tree}()$  is called to construct the 2-dimension induced tree, which is similar to the function  $\text{one-tree}()$ . After evaluating this function, the nodes and edges are selected, where  $V(T_n) \leftarrow \{u_0, u_1, u_2\}$  and  $E(T_n) \leftarrow E(T_n) \cup (u_0, u_2)$ . These nodes divided into bipartite partitions as  $V_1(T_n) \leftarrow \{u_0\}$  and  $V_2(T_n) \leftarrow \{u_1, u_2\}$ . If still  $\text{flag}! = 1$ , the  $n$ -dimension induced tree can be extended by the function  $\text{induced-tree}()$  until  $\text{flag} == 1$ . This function is similar to the function  $\text{one-tree}()$ . The new nodes are extended by the bipartite partitions, which is describe in Algorithm 2.

**4.3. Elimination stages.** Applying a dominating reducing rule can reduce some redundant nodes in the CDS. In this section, the elimination stage is designed to minimize the size of a CDS. If all the nodes of the

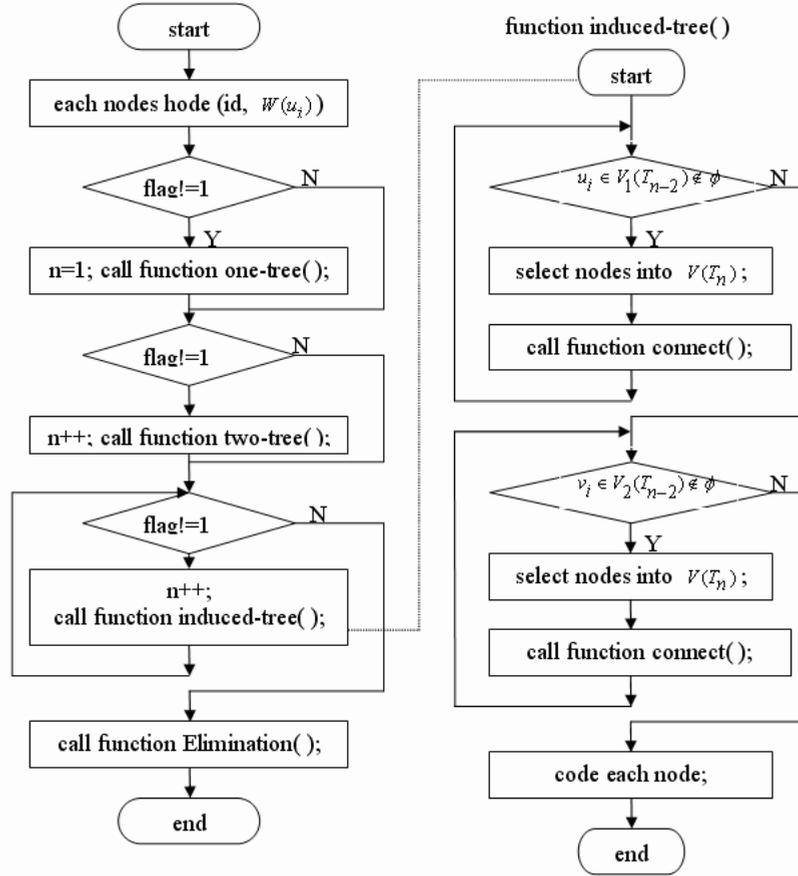


Fig. 4. Overall mechanism of CDS-ITCC- $G^*$ .

CDS, nodes' non-CDS neighbours and the node itself are covered by at least two nodes of the CDS, the node of the CDS is redundant and deleted from the CDS. The pruning stage is introduced in detail in Algorithm 3.

### 5. Theoretical analysis

This section will elaborately prove the following facts: (i) the correctness that all nodes with  $f(u_i) = 1$  or 2 form a CDS, (ii) calculation of the approximation ratio of the CDS-ITCC- $G^*$  algorithm, (iii) the cardinality of the induced trees is a Fibonacci sequence, and an upper bound of the number of the DS is established.

**Theorem 1.** Given a graph  $G = (V, E)$ , before the elimination stage, all nodes in  $T_n$  with  $f(u_i) = 1$  form an MIS.

*Proof.* The nodes  $u, v$  are connected in  $G^*$  if, and only if the nodes  $u, v$  are non-adjacent in graph  $G$ . At the initial stage,  $u_0$  is selected as an initiator, and  $f(u_0) = 1$ , after operating Line 2 in Algorithm 1, each node in  $N_G(u_0)$  is dominated by  $u_0$  and belongs to  $V_d$ . The algorithm is being completed until  $flag == 1$ . Lines 4, 8 and 12 in

Algorithm 1 are employed to construct the induced tree, each node  $v$  with  $f(v)=1$  in  $V(T_n)$  is selected from

$$N_{G^*}(v_i) \setminus (V_d \cup CDS),$$

so all nodes in  $T_n$  are independent, which either belong to  $N(T_n)$  or  $V_d$ . According to Definition 2,  $V'$  is an MIS when no independent vertex can be added into  $V'$ . If there is an independent vertex which can be added into  $T_n$ , there must be  $flag! = 1$  (Lines 3, 6 and 10 in Algorithm 1), which contradicts the fact that the algorithm is complete. Therefore, all nodes in  $T_n$  with  $f(u_i) = 1$  form an MIS. ■

**Theorem 2.** Given a graph  $G = (V, E)$ , all nodes with  $f(u_i) = 1$  or 2 form a CDS.

*Proof.* All nodes  $u_i$  with  $f(u_i) = 1$  form a DS. Now we need to show that the set of nodes with  $f(u_i) = 1$  or 2 is connected. Assume that  $u, v \in DS$  are two adjacent nodes in  $G^*$ ; after calling the function connect (), the relay node  $w$  with  $f(w) = 2$  is selected from  $N_G(u) \cap N_G(v)$ , where  $w$  is the common neighbour of  $u, v$ . If there is no such node, the distance between  $u$  and  $v$  is more than two hops, neither  $u$  and  $v$  are not adjacent in  $G^*$  according to

**Algorithm 2.** Induced-tree().

**Require:**  $V(T_n)$

- 1: for each  $u_i \in V_1(T_{n-2}) \notin \phi$  and  $\text{flag}! = 1$  and  $N_{G^*}(u_i) \setminus (V_d \cup CDS) \neq \phi$
- 2:      $i = 0$ ;
- 3:      $V(T_n) = V(T_{n-2})$ ;
- 4:     select a node in  $N_{G^*}(u_i) \setminus (V_d \cup CDS)$ , with the maximum weight  $u_j$  with  $f(u_j) = 1$ ;
- 5:      $V(T_n) = V(T_n) \cup u_j$ ,  $E(T_n) = E(T_n) \cup (u_i, u_j)$ ;
- 6:      $i++$ ;
- 7:     call function  $\text{connect}(V(T_n)); \text{dom}(CDS)$ ;
- 8: end for;
- 9: for each  $v_i \in V_2(T_{n-2}) \notin \phi$  and  $\text{flag}! = 1$  and  $N_{G^*}(v_i) \setminus (V_d \cup CDS) \neq \phi$
- 10:     $i = 0$ ;
- 11:    select a node in  $N_{G^*}(v_i) \setminus (V_d \cup CDS)$ , with the maximum weight  $v_j$  with  $f(v_j) = 1$ ;
- 12:     $V(T_n) = V(T_n) \cup v_j$ ,  $E(T_n) = E(T_n) \cup (v_i, v_j)$ ;
- 13:    call function  $\text{connect}(V(T_n)); \text{dom}(CDS)$ ;
- 14:    if  $\text{flag}! = 1$
- 15:    select a node in  $N_{G^*}(v_i) \setminus (V_d \cup CDS)$ , with the maximum weight  $v_k$  with  $f(v_k) = 1$ ;
- 16:     $V(T_n) = V(T_n) \cup v_k$ ,  $E(T_n) = E(T_n) \cup (v_i, v_k)$ ;
- 17:    endif;
- 18:     $i++$ ;
- 19:    call function  $\text{connect}(V(T_n)); \text{dom}(CDS)$ ;
- 20: end for;
- 21: each  $u_i \in V_1(T_{n-2})$ ,  $u_i$  labelled by  $u_i00$ ;
- 22: each  $v_i \in V_2(T_{n-2})$ ,  $v_i$  labelled by  $v_i00$ ;
- 23: each  $u_j$  code of  $u_j01$ ,  $v_j$  labelled by  $v_j01$ , and  $v_k$  labelled by  $v_k10$ ;
- 24:  $V_1(T_n) = \{u_j01, v_i00 : u_j \in V_1(T_{n-2}), v_i \in V_2(T_{n-2})\}$ ,  $V_2(T_n) = V(T_n) - V_1(T_n)$ ;

**Algorithm 3.** Elimination().

**Require:**  $G_T(V(T_n), E(T_n)), CDS$

- 1: for each node  $u \in V(T_n)$
- 2:     if  $(\text{suff}(u)! = 00)$
- 3:         if  $(N_G[u] \subset N_G[CDS - u])$
- 4:              $V(T_n) = V(T_n) - u$ ,  $f(u) = 3$ ;
- 5:         endif;
- 6:     endif;
- 7: end for;

the Definition 4, a contradiction. Therefore, all nodes with  $f(u_i) = 1$  or 2 are connected.

After the elimination stage, the dominators in the boundary, whose neighbourhood includes themselves, are dominated by other nodes in the CDS,  $f(u_i) = 3$  (Lines 2, 3 and 4 in Algorithm 3). It does not affect the connectivity

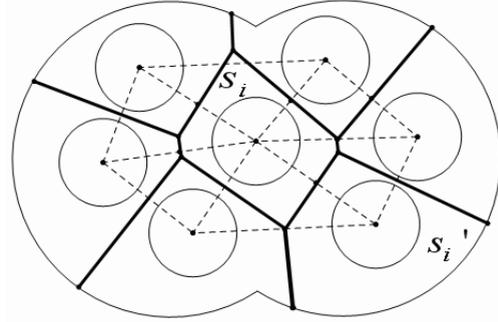


Fig. 5. One example for the Voronoi diagram.

and the domination. Therefore, all nodes with  $f(u_i) = 1$  or 2 form a CDS. ■

**Theorem 3.** Given a graph  $G = (V, E)$ , let  $\alpha$  be the DS obtained from the algorithm CDS-ITCC- $G^*$ . Then

$$\alpha \leq 3.2833\gamma + 4.5590 \quad (2)$$

where  $\gamma$  is the size of any optimal CDS of  $G$ .

*Proof.* The algorithm CDS-ITCC- $G^*$  is based on constructing the MIS. Let  $|\text{MIS}| = \alpha$ , and  $|\text{MCDS}| = \gamma$ . There is a close relationship between  $\alpha$  and  $\gamma$  in the unit disk graph. A new graph  $G'$  can be constructed by increasing the transmission range of nodes from 1 to 1.5, which are in the MCDS, and decreasing the transmission range of the rest nodes from 1 to 0.5, so that the cover area of the nodes in  $V$  is located inside the area formed by the MCDS. If we select an MIS for  $G$ , for the nodes in the the MIS, the distance between any two nodes' cover area should be greater than 1. Since the transmission range of nodes in  $V \setminus \text{MCDS}$  for  $G'$  is decreased to 0.5, any two of cover areas from the MIS will not intersect each other. Now we want to find the rough bound between the sum of the maximum area of the MIS and the area of the MCDS:  $\alpha/\gamma$ .

For a graph  $G'$ , for each cover area of each node  $u_i \in \text{MIS}$ , as shown in Fig. 5, the corresponding Voronoi cell, with the outer boundary, is the boundary for the MCDS. Clearly, each non-boundary Voronoi cell is a convex polygon, and the boundary Voronoi cells is some special kind of polygons with one arc edge. With exactly  $i$  edges, the number of non-boundary Voronoi cells are denoted as  $\alpha_i$ , the boundary Voronoi cells are denoted as  $\alpha'_i$ . The  $i$ -polygon cell, which is the minimum area of the non-boundary cell, is denoted as  $s_i$ , and the minimum area of the boundary cell is denoted as  $s'_i$ . The following formula (Gao et al., 2009) gives a bound for the approximation ratio  $\alpha/\gamma$ :

$$\begin{aligned}
 S_{MIS} &= \sum_i (s_i \alpha_i + s'_i \alpha'_i) \\
 &= 1.299\alpha_3 + 1.178\alpha'_3 \\
 &\quad + \alpha_4 + 0.972\alpha'_4 + 0.9082\alpha_5 \\
 &\quad + 0.8968\alpha'_5 + 0.866(\alpha_6 + \dots) \\
 &\quad + 0.8546\alpha'_6 + 0.8525(\alpha'_7 + \dots) \\
 &\leq S_{MCDS} \leq 2.9435\gamma + 4.1251.
 \end{aligned} \tag{3}$$

For an induced tree  $T_k$ , it is easy to see that the vertex  $v$  has the maximum degree in  $G^*$ , where  $d_{G^*}(v) = k - 1$ . The node is in the intermediate cell with the Voronoi division, as to simplify the calculation. The Voronoi division will be modified as regularization, such that the degree of any vertex of  $v$  in the Voronoi division is exactly  $k$ . For any vertex  $v$  whose  $d(v) = d < k$ , add  $k - d$  new vertices  $v_1, v_2, \dots, v_{k-d}$  as its new neighbours in clockwise ordering, such that the distance between  $v$  and any  $v_i$  is sufficiently small, so that the area of all Voronoi cells will almost remain the same, and the number of edges of original Voronoi cells is no more than that of new cells. By using Euler's formula, we get

$$\sum_i (\alpha_i + \alpha'_i) - m + n = 1. \tag{4}$$

Since  $G'$  is a  $k$ -regularization graph, according to the principle of the shaking hand, there is  $2m = kn$ . Then

$$\sum_i (\alpha_i + \alpha'_i) - \frac{k-2}{2}n = 1. \tag{5}$$

Each edge is exactly in two faces and there is at least one edge belonging to the outer face for any boundary cell. Hence

$$\sum_i i\alpha_i + \sum_i (i+1)\alpha'_i - kn \leq 0. \tag{6}$$

By the formulae (5) and (6), we have

$$\left(\frac{2k}{k-2} - 3\right)\alpha_3 + \left(\frac{2k}{k-2} - 4\right)\alpha'_3 + \dots \geq \frac{2k}{k-2}. \tag{7}$$

By the formulae (3) and (7), we have

$$\begin{aligned}
 &\dots + [0.8661 - \left(\frac{2k}{k-2} - 6\right) \times 0.0114]\alpha_6 + \dots \\
 &\quad + [0.8546 - \left(\frac{2k}{k-2} - 7\right) \times 0.0114]\alpha'_6 + \dots \\
 &\leq 2.9435\gamma + 4.1251 - 0.0114 \times \frac{2k}{k-2}.
 \end{aligned} \tag{8}$$

From  $\alpha = \sum_i (\alpha_i + \alpha'_i)$ , we have

$$\alpha \leq \frac{2.9435\gamma + 4.1251 - 0.0114 \times \frac{2k}{k-2}}{0.8661 - \left(\frac{2k}{k-2} - 6\right) \times 0.0114}. \tag{9}$$

Since Wu *et al.* (2010) proved that a dominating disk can contain at most five independent nodes, set  $k = 5$ . Then  $\alpha \leq 3.2833\gamma + 4.5590$ . ■

**Theorem 4.** Given a graph  $G = (V, E)$ , the number of nodes in the DS based on the induced tree of the crossed cube is approximately equal to

$$0.447 \times [1.618^{n+2} - (-0.618)^{n+2}], \tag{10}$$

where  $n \geq 1$  is the dimensional of the crossed cube.

*Proof.* According to Definition 8, the number of the nodes in the induced tree consists of bipartite partition  $V_1(T_n)$  and  $V_2(T_n)$ . We have

$$|T_n| = |V_1(T_n)| + |V_2(T_n)|. \tag{11}$$

Especially,  $|T_1| = 2$  in  $CQ_1$ ,  $|T_2| = 3$  in  $CQ_2$ , and  $|T_3| = 5$  in  $CQ_3$ .

According to the definition of the induced trees of the crossed cube,  $|T_n|$  is a Fibonacci series for  $n = 1, 2, \dots$  satisfying

$$|T_n| = |T_{n-1}| + |T_{n-2}| \quad (n \geq 3). \tag{12}$$

Assuming that

$$|T_n| = F_n, \tag{13}$$

and

$$F_{n+1} + xF_n = y(F_n + xF_{n-1}), \tag{14}$$

we have

$$F_{n+1} = (y-x)F_n + xyF_{n-1} \tag{15}$$

and

$$F_{n+1} = F_n + F_{n-1}. \tag{16}$$

Hence

$$\left(x = \frac{\sqrt{5}-1}{2}, y = \frac{\sqrt{5}+1}{2}\right)$$

or

$$\left(x = \frac{-\sqrt{5}-1}{2}, y = \frac{-\sqrt{5}+1}{2}\right).$$

Take the former solution; then

$$\begin{aligned}
 F_{n+1} + \frac{\sqrt{5}-1}{2}F_n \\
 = \frac{\sqrt{5}+1}{2}(F_n + \frac{\sqrt{5}-1}{2}F_{n-1}).
 \end{aligned} \tag{17}$$

Assume that

$$b_n = F_{n+1} + \frac{\sqrt{5}-1}{2}F_n. \tag{18}$$

Then

$$b_n = \frac{\sqrt{5}+1}{2}b_{n-1},$$

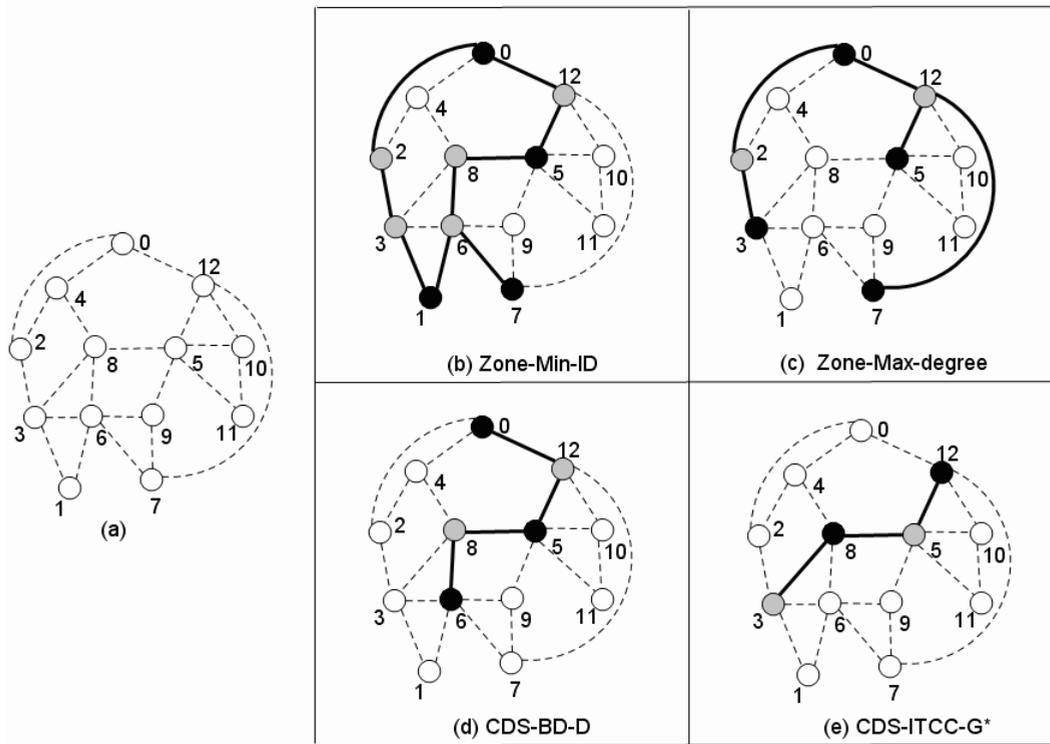


Fig. 6. One comparative example of the number of MCDSs.

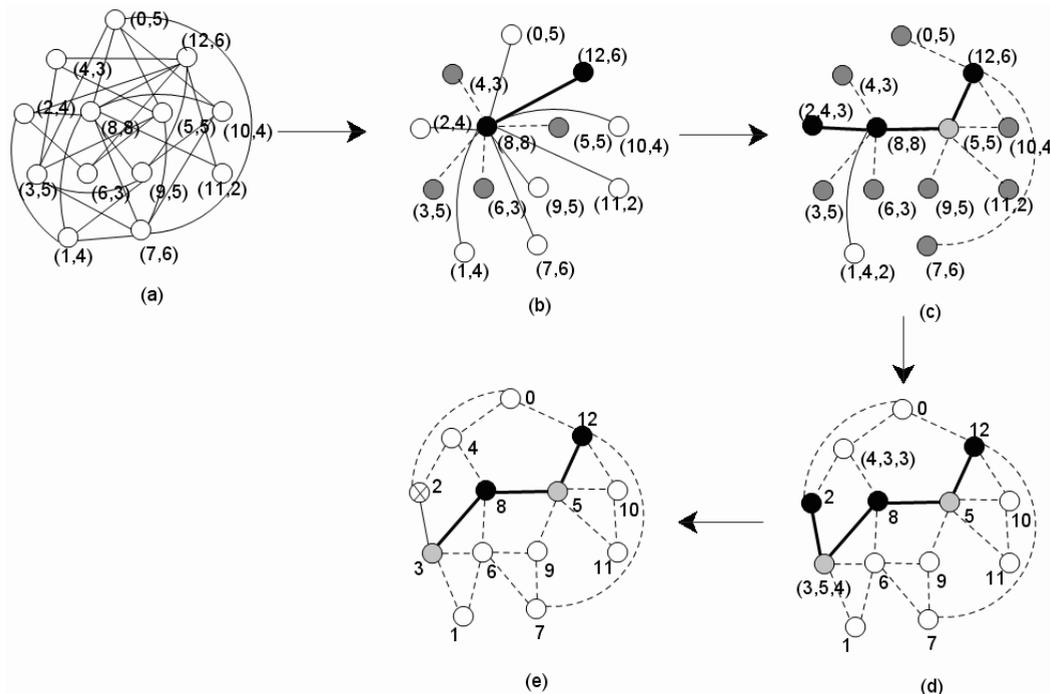


Fig. 7. Construction of a CDS by CDS-ITCC-G\*.

so the series  $b_n$  is geometric, with the first element

$$b_1 = \left(\frac{\sqrt{5}+1}{2}\right)^3,$$

and the common ratio

$$q = \frac{\sqrt{5}+1}{2},$$

so

$$b_n = \left(\frac{\sqrt{5}+1}{2}\right)^{n+2}.$$

Hence

$$F_{n+1} + \frac{\sqrt{5}-1}{2}F_n = \left(\frac{\sqrt{5}+1}{2}\right)^{n+2}. \quad (19)$$

Assume that

$$F_{n+1} + x\left(\frac{\sqrt{5}+1}{2}\right)^{n+3} = \frac{1-\sqrt{5}}{2}\left[F_n + x\left(\frac{\sqrt{5}+1}{2}\right)^{n+2}\right]. \quad (20)$$

By the formulae (19) and (20), there is  $x = -1/\sqrt{5}$  such that

$$F_{n+1} - \frac{1}{\sqrt{5}}\left(\frac{\sqrt{5}+1}{2}\right)^{n+3} = \frac{1-\sqrt{5}}{2}\left[F_n - \frac{1}{\sqrt{5}}\left(\frac{\sqrt{5}+1}{2}\right)^{n+2}\right]. \quad (21)$$

Assume that

$$c_n = F_n - \frac{1}{\sqrt{5}}\left(\frac{\sqrt{5}+1}{2}\right)^{n+2}.$$

The series  $c_n$  is a geometric one with the first element

$$c_1 = 3 - \frac{1}{\sqrt{5}}\left(\frac{\sqrt{5}+1}{2}\right)^3 = -\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^3 \quad (22)$$

and the common ratio  $(1-\sqrt{5})/2$  so

$$c_n = -\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+2}.$$

We have

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{\sqrt{5}+1}{2}\right)^{n+2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+2}\right] \approx 0.447 \times [1.618^{n+2} - (-0.618)^{n+2}]. \quad (23)$$

Therefore the number of nodes in the DS based on the induced tree of the crossed cube is approximately equal to  $0.447 \times [1.618^{n+2} - (-0.618)^{n+2}]$ . ■

Theorem 4 illustrates that for the network, when the dimensionality of the induced tree of the crossed cube has been settled, the number of the nodes in CDS can be calculated, which is more accurate than the approximation ratio.

## 6. Simulations

### 6.1. One example of the CDS-ITCC- $G^*$ algorithm.

The performance of the CDS-ITCC- $G^*$  algorithm is evaluated by comparing it with some traditional CDS algorithms (Kim *et al.*, 2009; Han, 2009) in this section. In order to illustrate how CDS-ITCC- $G^*$  outperforms these CDS clustering algorithms in terms of reducing the CDS size, the same distribution of nodes are used. Figure 6 shows a comparative example quoted from (Bahaa-Eldin *et al.*, 2012). In Figs. 6(b)–(e), each CDS is constructed according to a certain CDS algorithm. Figures 6(b)–(d) show the CDS when Zone-Min-ID (Han, 2009), Zone-Max-degree (Han, 2009) and CDS-BD-D (Kim *et al.*, 2009) algorithms are applied, the number of the CDS being 9, 6 and 5, respectively. It is shown that the CDS-ITCC- $G^*$  algorithm has the smallest CDS size of 4 which is shown in Fig. 6(e).

Figures 7(a)–(e) illustrate the execution stages of our algorithm. At first the graph  $G^*$  is constructed in Fig. 7(a). Node 8 has the maximum degree in  $G^*$ , so it is selected as the initiator coloured with black. Let  $u_0 = 8$ , and labelled by 0,  $f(8) = 1$  (Line 2 in Alg.1). Call the function `dom()` in Fig. 7(b). It is easy to see that the nodes 3, 4, 5 and 6 are dominated by the node 8. However,  $\{8, 3, 4, 5, 6\} \neq V$ , so the algorithm continues as specified (Line 3 in Alg. 1). Then call the function `one-tree()` (Line 4 in Alg. 1).

Node 12 has the maximum weight in  $N_{G^*}(8)$ , as it has the greatest degree  $d_{G^*}(12) = 6$ . Node 12 is selected into the induced tree coloured with black. Let  $u_1 = 12$ , and labelled by 1,  $f(12) = 1$ , and  $V(T_n) = \{u_0, u_1\}$ . Then call the function `connect( $V(T_n)$ )`. There is only one node 5 satisfying  $w_i \in N_G(8) \cap N_G(12)$ , so  $CDS = V(T_n) \cup \{5\}$  and  $f(5) = 2$  coloured with gray as shown in Fig. 7(c). Since  $CDS \cup V_d \neq V$ , call the function `two-tree( $V(T_n)$ )` (Line 8 in Alg. 1).

Since the label of each node in the 2-dimensional crossed cube is 2-bit, node  $u_0$  is labelled by 00, node  $u_1$  labelled by 01, and there are two nodes

$$\{1, 2\} \subseteq N_{G^*}(8) \setminus (CDS \cup V_d).$$

Node 2 is selected into the induced tree coloured with black as  $d_{G^*}(1) = d_{G^*}(2)$  and  $d_G(1) < d_G(2)$ . Let  $u_2 = 2$ , and labelled by 10,  $f(2) = 1$ , and  $V(T_n) = \{u_0, u_1, u_2\}$ . In Fig. 7 (d), call the function `connect( $V(T_n)$ )`. There are two nodes satisfying  $\{3, 4\} \subseteq N_G(8) \cap N_G(2)$ . Node 3 is selected as the connector node coloured with grey as  $d_{G^*}(3) > d_{G^*}(4)$ , so  $f(3) = 2$  and  $CDS = V(T_n) \cup \{3\}$ . Now `flag == 1`, so execute Alg. 3, in Fig. 7(e) according to the elimination stage  $N_G[2] \subset N_G[CDS - \{2\}]$ , so  $V(T_n) = V(T_n) - \{2\}$ ,  $f(2) = 3$  (Line 3 in Alg. 3). The CDS consists of nodes 3, 5, 8 and 12 as shown in Fig. 6(e).

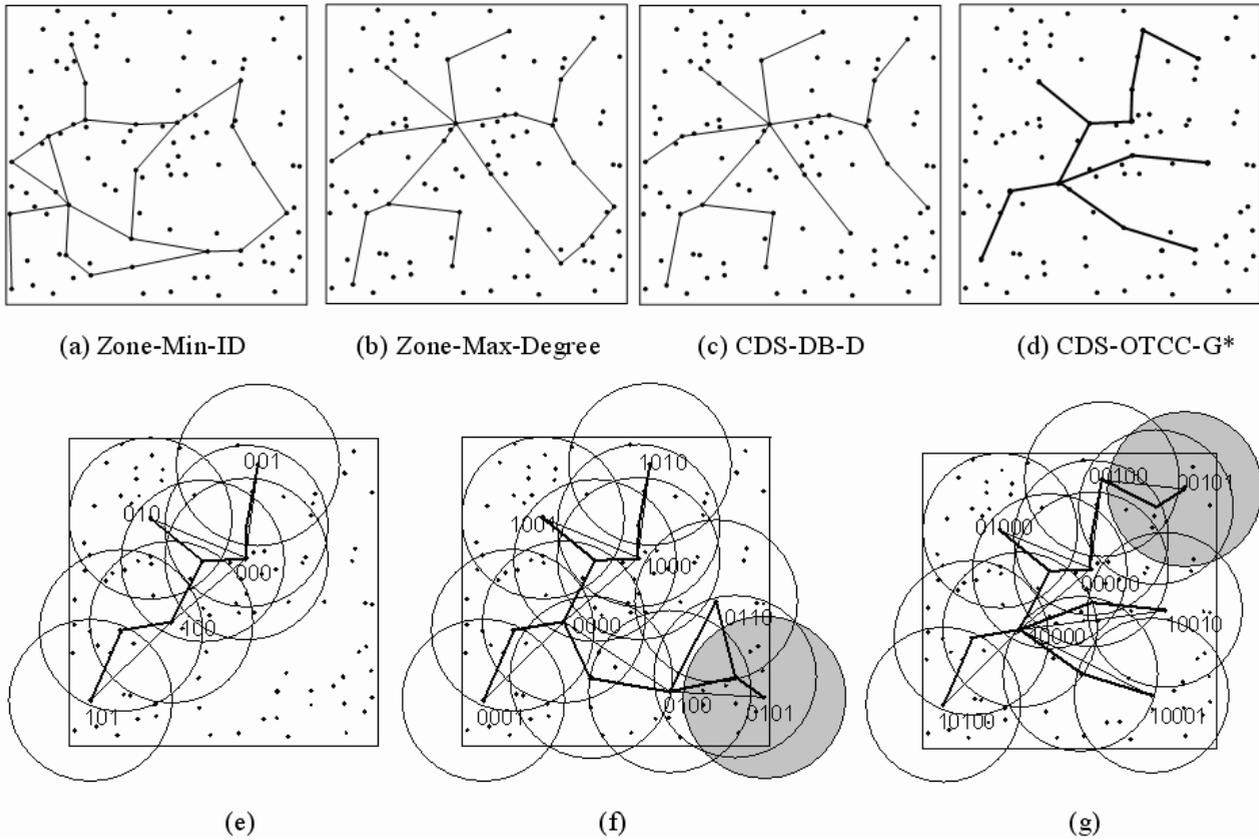


Fig. 8. Comparative example.

**6.2. Comparative example.** Another comparative example also quoted from the work of Bahaa-Eldin *et al.* (2012) is presented in Figs. 8(a)–(d). Network parameters used in these figures are identical, 100 nodes are in a  $100 \times 100$  network generated randomly with 25 as the transmission range. Figures 8(a)–(c) show the CDS when Zone-Min-ID, Zone-Max-degree and CDS-BD-D algorithms are applied and the CDS sizes are 24, 23 and 21, respectively. The induced tree  $T_3$  has been structured in Fig. 8(e), and  $CDS \cup V_d \neq V$ . Consequently, the algorithm must be continued. The induced tree  $T_4$  has been structured in Fig. 8(f), where

$$V(T_n) = \{0000, 0001, 1000, 1001, 1010, 0101, 0110\}.$$

Obviously,  $CDS \cup V_d \neq V$ .

Another  $N_G[0101] \subset N_G[CDS - 0101]$ , so  $\{0101\}$  is eliminated. Continue the algorithm. The induced tree  $T_5$  has been structured in Fig. 8(g). We have

$$N_G[00101] \subset N_G[CDS - \{00101\}],$$

so  $\{00101\}$  is eliminated. After all nodes have determined their states by CDS-ITCC- $G^*$  the total size of the CDS is 13 nodes as shown in Fig. 8(d). It also can show that CDS-ITCC- $G^*$  has the smallest CDS size.

In order to fully evaluate the performance of CDS size, we will complete our CDS-ITCC- $G^*$  algorithm with the recent ones, CDS-BD-D (Kim *et al.*, 2009) and E-MCDS (Tang *et al.*, 2012). The nodes are randomly distributed in the network, and all points in the figures are simulated 100 times. Consider two system parameters, the number of nodes in the space and the common transmission range of nodes. The number of nodes is increased by 10 from 10 to 100, and the maximum transmission range varies between 20 and 30. Figure 9 shows the comparison of the algorithms in terms of CDS size.

From Fig. 9, it is easy to see that the CDS size of CDS-ITCC- $G^*$  is the smallest among all the compared algorithms, the CDS size of CDS-BD-D being the biggest. The size of CDS is increasing as the number of nodes becomes bigger. Compared with (a) and (b), it is easy to see that the CDS size becomes smaller as the transmission range increases, which is because the transmission range is bigger, the covered area is larger, and at the same time the network area size is finite and then the size of CDS is smaller. From these figures it is evident that CDS-ITCC- $G^*$  provides the smallest CDS size.

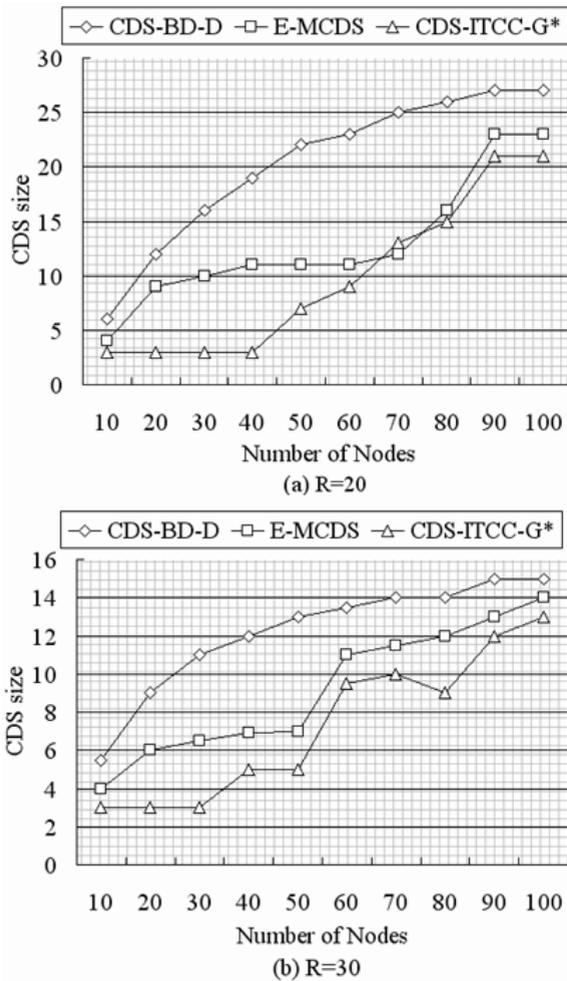


Fig. 9. CDS size versus the number of nodes  $R = 20$  (a),  $R = 30$  (b).

## 7. Conclusion

In this paper, a new induced tree based on the crossed cube is defined and constructed recursively. A novel algorithm CDS-ITCC- $G^*$  based on the induced tree in the  $G^*$  is presented, which can generate a maximal independent set with the approximation ratio of  $3.2833\gamma + 4.5590$ , where  $\gamma$  is the size of any optimal CDS. It was proven that the cardinality of the induced trees is a Fibonacci sequence, an upper bound of the number of the DS is  $0.447 \times [1.618^{n+2} - (-0.618)^{n+2}]$ . Compared with some traditional algorithms, the CDS-ITCC- $G^*$  algorithm enjoys the smallest CDS size.

Our interest for further investigation is to find some methods to construct  $k$ -connected  $m$ -dominating sets for fault tolerance, based on induced trees of the crossed cube. Besides, we are interested in designing new distributed CDS construction algorithms considering more factors such as load balance, fault tolerance, and so forth.

## Acknowledgment

The authors wish to thank the National Natural Science Foundation of China (grants no. 61072080 and U1405255), the development project of Fujian provincial strategic emerging industries technologies: *Key Technologies in Development of Next Generation Integrated High Performance Gateway*, Fujian development and reform commission high-technical [2013]266, the Natural Science Foundation of Fujian Province (nos. 2013J01221, 2013J01222, J01223), the Education Department of Fujian Province science and technology project (JA13215, JA12073).

## References

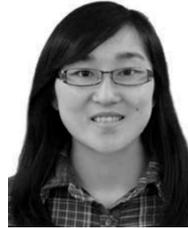
- Bahaa-Eldin, A.M., Hassan, D.S.M. and Fahmy, H. (2012). RCA: Efficient connected dominated clustering algorithm for mobile ad hoc networks, *arXiv preprint*, arXiv:1211.0673.
- Cheng, B., Fan, J., Jia, X. and Wang, J. (2013). Dimension-adjacent trees and parallel construction of independent spanning trees on crossed cubes, *Journal of Parallel and Distributed Computing* 73(5): 641–652.
- Cheng, X., Huang, X., Li, D., Wu, W. and Du, D.Z. (2003). A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks, *Networks* 42(4): 202–208.
- Dai, F. and Wu, J. (2005). On constructing  $k$ -connected  $k$ -dominating set in wireless networks, *19th IEEE International Parallel and Distributed Processing Symposium, Denver, CO, USA*, p. 81a.
- Ding, L., Wu, W., Willson, J., Du, H., Lee, W. and Du, D.Z. (2011). Efficient algorithms for topology control problem with routing cost constraints in wireless networks, *IEEE Transactions on Parallel and Distributed Systems* 22(10): 1601–1609.
- Ding, L., Wu, W., Willson, J., Du, H. and Lee, W. (2012). Efficient virtual backbone construction with routing cost constraint in wireless networks using directional antennas, *IEEE Transactions on Mobile Computing* 11(7): 1102–1112.
- Du, H., Ye, Q., Wu, W., Lee, W., Li, D. and Du, D. (2011). Constant approximation for virtual backbone construction with guaranteed routing cost in wireless sensor networks, *INFOCOM, Shanghai, China*, pp. 1737–1744.
- Funke, S., Kesselman, A., Meyer, U. and Segal, M. (2006). A simple improved distributed algorithm for minimum cds in unit disk graphs, *ACM Transactions on Sensor Networks* 2(3): 444–453.
- Gao, X., Wang, Y., Li, X. and Wu, W. (2009). Analysis on theoretical bounds for approximating dominating set problems, *Discrete Mathematics, Algorithms and Applications* 1(1): 71–84.
- Goli, J.D. (2012). A new authentication model for ad hoc networks, *International Journal of Information Security* 11(5): 333–347.

- Han, B. (2009). Zone-based virtual backbone formation in wireless ad hoc networks, *Ad Hoc Networks* **7**(1): 183–200.
- He, J., Ji, S., Pan, Y. and Cai, Z. (2013). Approximation algorithms for load-balanced virtual backbone construction in wireless sensor networks, *Theoretical Computer Science* **507**: 2–16.
- Kim, D., Wang, W., Li, X., Zhang, Z. and Wu, W. (2010). A new constant factor approximation for computing 3-connected m-dominating sets in homogeneous wireless networks, *INFOCOM, San Diego, CA, USA*, pp. 1–9.
- Kim, D., Wu, Y., Li, Y., Zou, F. and Du, D.Z. (2009). Constructing minimum connected dominating sets with bounded diameters in wireless networks, *IEEE Transactions on Parallel and Distributed Systems* **20**(2): 147–157.
- Li, M., Wan, P.J. and Yao, F. (2011). Tighter approximation bounds for minimum cds in unit disk graphs, *Algorithmica* **61**(4): 1000–1021.
- Li, D., Kim, D., Zhu, Q., Liu, L. and Wu, W. (2012a). Minimum total communication power connected dominating set in wireless networks, in X. Wang, R. Zheng, T. Jing and X. Kai (Eds.), *Wireless Algorithms, Systems, and Applications*, Springer, Berlin/Heidelberg, pp. 132–141.
- Li, Y., Wu, Y., Ai, C. and Beyah, R. (2012b). On the construction of k-connected m-dominating sets in wireless networks, *Journal of combinatorial optimization* **23**(1): 118–139.
- Liao, Q. and Li, Z. (2013). Portfolio optimization of computer and mobile botnets, *International Journal of Information Security* **13**(1): 1–14.
- Lin, Z., Xu, Wang, D. and Gao, J. (2006). A coloring based backbone construction algorithm in wireless ad hoc network, in Y.-C. Chung and J.E. Moreira (Eds.), *Advances in Grid and Pervasive Computing*, Springer, Berlin/Heidelberg, pp. 509–516.
- Liu, Q., Zhang, Z., Hong, Y., Wu, W. and Du, D.Z. (2013). A PTAS for weak minimum routing cost connected dominating set of unit disk graph, in A. Chinchuluun, P.M. Pardalos, R. Enkhbat and E.N. Pistikopoulos (Eds.), *Optimization, Simulation, and Control*, Springer, New York, NY, pp. 131–142.
- Misra, R. and Mandal, C. (2010). Minimum connected dominating set using a collaborative cover heuristic for ad hoc sensor networks, *IEEE Transactions on Parallel and Distributed Systems* **21**(3): 292–302.
- Padmavathy, M.C. (2010). Performance evaluation of energy efficient modulation scheme and hop distance estimation for WSN, *International Journal of Communication Networks and Information Security* **2**(1): 44–49.
- Tang, Q., Yang, K., Li, P., Zhang, J., Luo, Y. and Xiong, B. (2012). An energy efficient MCDS construction algorithm for wireless sensor networks, *EURASIP Journal on Wireless Communications and Networking* **2012**: 83.
- Thai, M.T., Zhang, N., Tiwari, R. and Xu, X. (2007). On approximation algorithms of k-connected m-dominating sets in disk graphs, *Theoretical Computer Science* **385**(1): 49–59.
- Wan, P.J., Alzoubi, K.M. and Frieder, O. (2002). Distributed construction of connected dominating set in wireless ad hoc networks, *INFOCOM, 21st Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY, USA*, Vol. 3, pp. 1597–1604.
- Wan, P.J., Wang, L. and Yao, F. (2008). Two-phased approximation algorithms for minimum CDS in wireless ad hoc networks, *28th International Conference on Distributed Computing Systems, Beijing, China*, pp. 337–344.
- Wang, F., Thai, M.T. and Du, D.Z. (2009). On the construction of 2-connected virtual backbone in wireless networks, *IEEE Transactions on Wireless Communications* **8**(3): 1230–1237.
- Wang, Y., Wang, X. and Wu, G. (2011). An equivalent definition of the crossed cube, *Chinese Quarterly Journal of Mathematics* **26**(2): 234–238.
- Wu, J. and Li, H. (1999). On calculating connected dominating set for efficient routing in ad hoc wireless networks, *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, WA, USA*, pp. 7–14.
- Wu, W., Du, H., Jia, X., Li, Y. and Huang S.C.H. (2006). Minimum connected dominating sets and maximal independent sets in unit disk graphs, *Theoretical Computer Science* **352**(1): 1–7.
- Wu, W., Gao, X., Pardalos, P.M. and Du, D.Z. (2010). Wireless networking, dominating and packing, *Optimization Letters* **4**(3): 347–358.
- Wu, Y. and Li, Y. (2008). Construction algorithms for k-connected m-dominating sets in wireless sensor networks, *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hong Kong SAR, China*, pp. 83–90.
- Wu, Y., Wang, F., Thai, M. T. and Li, Y. (2007). Constructing k-connected m-dominating sets in wireless sensor networks, *Military Communications Conference, MILCOM, Orlando FL, USA*, pp. 1–7.
- Xu, L. and Lin, Z. (2007). Graph coloring based minimal connected dominating set algorithm in wireless ad hoc networks, *Journal on Communications* **28**(3): 108–114.
- Zam, A. and Movahedinia, N. (2013). Performance improvement of cache management in cluster based manet, *International Journal of Computer Network and Information Security* **5**(10): 24–29.
- Zhao, Y., Wu, J., Li, F. and Lu, S. (2012). On maximizing the lifetime of wireless sensor networks using virtual backbone scheduling, *IEEE Transactions on Parallel and Distributed Systems* **23**(8): 1528–1535.
- Zhu, W.T., Xiang, Y., Zhou, J.D., Deng, R.H. and Bao, F. (2011). Secure localization with attack detection in wireless sensor networks, *International Journal of Information Security* **10**(3): 155–171.
- Zou, F., Wang, Y., Xu, X.H., Li, X., Du, H., Wan, P. and Wu, W. (2011). New approximations for minimum-weighted dominating sets and minimum-weighted connected

dominating sets on unit disk graphs, *Theoretical Computer Science* **412**(3): 198–208.



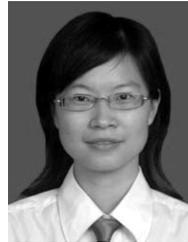
**Jing Zhang** received her B.Sc. and M.Sc. degrees at the School of Mathematics and Computer Science from Fujian Normal University, China, in 2004 and 2007, respectively. She is currently working toward her Ph.D. there. She is a lecturer at the School of Information Science and Engineering, Fujian University of Technology. Her research interests include network algorithms, network and information security.



**Wei Wu** received her Ph.D. from the University of Wollongong, Australia, in 2011. She is currently an associate professor in the School of Mathematics and Computer Science, Fujian Normal University, China. Her research interests include new public key cryptography systems and secure server-aided computation. She has published more than 20 referred research papers at international conferences and journals.



**Li Xu** is a professor and doctoral supervisor at the School of Mathematics and Computer Science at Fujian Normal University. He received his B.Sc. and M.Sc. degrees from that university in 1992 and 2001. He received the Ph.D. degree from the Nanjing University of Posts and Telecommunications in 2004. His scientific interests include network optimization and network security, complex networks and systems, wireless communication networks and communication, etc. Professor Xu has been invited to act as a PC chair or member at more than 30 international conferences. He is a member of the IEEE and ACM, and a senior member of the CCF and CIE in China. He has published over 100 papers in refereed journals and conferences.



**Xiucui Ye** received her B.Sc. and M.Sc. degrees from Fujian Normal University in 2006 and 2009, respectively. She is currently working toward her Ph.D. at the Graduate School of Systems and Information Engineering, University of Tsukuba, Japan. Her research interests are in data coding, fault tolerance and distributed storage in wireless networks.

Received: 3 January 2014  
Revised: 12 May 2014



**Shuming Zhou** received the Ph.D. degree in mathematics from Xiamen University, Xiamen, People's Republic of China, in 2005. He is currently an associate professor in the School of Mathematics and Computer Science, Fujian Normal University, Fuzhou, People's Republic of China. Dr. Zhou is a member of the Computer Society of China. His research interests include discrete optimization, network reliability, and fault-tolerant computing.