Backbone formulation algorithm in wireless sensor network based on cross-entropy method *

SHI Weiren[1], JIANG Yisong[1], ZHAO Ying[1]

(*College of Automation, Chongqing University, Chongqing* 400044, *China*)

Abstract: In wireless sensor network, virtual backbone is a cost effective broadcasting method. Connected dominating set formation is proposed to construct a virtual backbone. However, it is NP-Hard to find a minimum connected dominating set in an arbitrary graph. In this paper, based on cross-entropy method, we present a novel backbone formulation algorithm (BFA-CE) in wireless sensor network. In BFA-CE, a maximal independent set is got at first and nodes in the independent set are required to get their action sets. Based on those action sets, a backbone is generated with the cross-entropy method. Simulation results show that our algorithm can effectively reduce the size of backbone network within a reasonable message overhead, and it has lower average node degree. This approach can be potentially used in designing efficient broadcasting strategy or working as a backup routing of wireless sensor network.

Key words: wireless sensor network; backbone; connected dominated set; cross-entropy method

1. Introduction

A wireless sensor network (WSN) is composed by many sensor nodes and one or multiple sinks. In WSN, sensors gather data from the sensing environment and transfer those data to the sink nodes or base stations. Generally, WSNs are deployed in some emergent or temporary situations, such as managing energy plants, logistics and inventory, battlefields, and medical monitoring[1]. Broadcasting is a fundamental means in WSNs. Flooding is the most straightforward broadcasting way that widely used in most routing algorithm. However, there are many redundant retransmissions in the flooding and those redundant retransmissions may cause the broadcast storm problem. A broadcasting method based on Virtual backbone is an alternative to the broadcasting because only the nodes in the backbone are required to retransmit the broadcasting packages. On the other hand, a formulation that based on minimum connected dominating set (MCDS) is a graph-based promising approach to construct a backbone that has received many attentions[2-10]. However, it is NP-hard to get an MCDS of an attribute graph[11].

In this paper, we model a WSN as an undirected graph G(V, E). Where vertices set V represents the nodes in WSN, and E represent the links between distinct nodes of V. We also assume that every node in WSN has the same communication range. Thus, G is also a unit disk graph (UDG). In the following, we give some definitions that used in this paper (in those definitions, S is a subset of V).

Definition 1. In the graph G(V, E), S is a dominating set (DS) if and only if each vertex which is not in set will be joined to at least one member of the set by an edge.

Definition 2. In the graph G(V, E), S is a connected dominating set (CDS) if and only if S is a DS that induces a connected subgraph of G.

A MCDS is a connected dominating set with the smallest size among all possible connected dominating sets of G.

Definition 3. In the graph G(V, E), S is a maximal independent set (MIS) if and only if every

* This work is supported partially by the science and technology project of CQ CSTC (No. cstc2012jjA40037).

edge in G has at least one endpoint not in S and every vertex not in S has at least one neighbor in the set.

In [3], Alzoubi et al. proposed two MIS-based backbone formulation algorithms. The first algorithm requires a spanning tree to complete the process that constructs a CDS. The second algorithm does not need the spanning tree and enables the maintenance of the weakly-connected dominating set to be simpler. In [4], another MIS-based backbone formulation is proposed by Li et al. This algorithm consists of two stages. The first one constructs a MIS of the WSN, and the second one connects those nodes in the MIS by a Steiner tree. A prune-based algorithm that prunes some redundant nodes from an original CDS based on "Rule 1" and "Rule 2" is presented by Dai[5]. In [6], Rule k, which is the enhanced method of the former method, is also mentioned by Dai and Wu. Another similar algorithm that purposed by Butenko is presented in [7]. Akbari Torkestani and Meybodi[8] described an intelligent CDS-based backbone formation algorithm in which the learning automata are used to construct the CDS of the network. Akbari also purposes another backbone formation algorithm[9] that considers the energy efficient of WSNs. In this algorithm, a learning automata-based heuristic is purposed to find an optimal solution of the proxy equivalent constrained CDS problem. In [10], a load-balanced virtual backbone construction algorithm is proposed by He et al. The size and the load-balance factors are introduced in He's algorithm when constructing the backbone of WSN. Bo[11] presents a zone-based algorithm, in which every node is assigned "Zone" and "Level". In Bo's algorithm, a dominating tree is constructed for each zone and some connectors are inserted to the backbone to connect adjacent zones.

In this paper, based on the cross-entropy method (CEM), we describe a method to construct a CDS in detail. We call this method a backbone formulation algorithm based on CEM (BFA-CEM). BFA-CEM consists of three stages. At the first stage, a MIS is got according to the ranks of nodes. At the second stage, each dominator gets its action set. Based on those action sets, a CDS will be got by sink with the CEM in the last stage. In BFA-CEM, the value of threshold γ and the values in the probability matrix are changed with time. That increases the probability of getting an approximation solution of MCDS and decrease the number of samples that may be large in the Monte Carlo sampling methods. The rest of this paper is organized as follows. In Section 2, we present relevant definitions of the problem. Section 3 reviews the cross-entropy method and our algorithm is shown in Section 4. Section 5 analyses the performance of BFA-CEM. The results of computational experiments are presented in Section 6. Finally, concluding remarks are given in Section 7.

2 Problem definitions

As shown before, the first stage of BFA-CEM is getting a MIS. In this paper, a simple and distributed method that described in [12] is used to get the MIS. The main objective of our algorithm is to find some connector to connect those nodes in the MIS. Let $B' = \{n_1, n_2, \cdots, n_m\}$ denotes the MIS that need to connect. χ is the set of CDSs includes all nodes in B'. X^* is the optimal solution, if

$$X^* = \arg\min_{X \in \chi} \{M(X)\} \tag{1}$$

where, X is a CDS in χ, M(X) is the number of nodes in X. Let $\gamma*$ is the size of X^*, that is

$$\gamma^* = \min_{X \in \chi} \{M(x)\} \tag{2}$$

In this paper, we get a CDS by sampling. Let $\varphi(\cdot)$ is a real-valued function on M(X) and it defines as

$$\varphi[M(x)] = I_{\{M(x)<\gamma\}} \tag{3}$$

Where γ is a positive number, called threshold, and

$$I_{\{M(x)<\gamma\}} = \begin{cases} 1 & M(x) < \gamma \\ 0 & others \end{cases} \tag{4}$$

Let ℓ be the expected of the stochastic event $\{M(x)<\gamma\}$ given in the form

$$\ell = P_f(M(X)<\gamma) = E_f[\varphi[M(x)]] = \int_{\chi} \varphi(M(X)) f(x) dx \quad (5)$$

Where, $f(x)$ is a probability density function (pdf). The value of ℓ also can be treated as the probability of $\{M(X)<\gamma\}$ because of the definition in (4). If ℓ is a small number, i.e. $<10^{-4}$, $\{M(X)<\gamma\}$ is called as a rare even. In this paper, the value of γ, which is the size of the CDS that we want to get, is close to $\gamma*$, and $\{M(X)<\gamma\}$ may be a rare event. If a crude Monte Carlo estimator is used, we should generate a plenty of samples to get an approximation solution of MCDS. To overcome this problem, CEM[13] is used in our algorithm, in which the probability density function is changed with time and increase the probability of $\{M(X)<\gamma\}$. Therefore, the number of samples is decreased.

3 Cross-entropy method

In CEM, the importance sampling (IS) is used. Importance sampling is a general technique for estimating properties of a particular distribution, while only having samples generated from a different distribution rather than the distribution of interest. As shown in (6), in importance sampling, another pdf g(x), which is called the importance sampling pdf, is intorduced.

$$\ell = \int \varphi[M(x)] \frac{f(x)}{g(x)} g(x) dx = E_g\left[\varphi[M(X)] \frac{f(X)}{g(X)}\right] \quad (6)$$

An estimator of ℓ is

$$\bar{\ell} = \frac{1}{N}\sum_{i=1}^{N}\left[\varphi[M(X_i)] \frac{f(X_i)}{g(X_i)}\right] \quad (7)$$

Where $X_1, \cdots, X_N$ are random samples from g(x). The choice of g(x) is important to the variance of (7). Rubinstein[13] has considered the next problem of minimizing the variance of $\bar{\ell}$ with respect to the g(x), that is

$$\min_g Var_g\left\{\varphi[M(X)] \frac{f(X)}{g(X)}\right\} \quad (8)$$

According to [13], an solution of (8) is $g*(x)$ ($Var_{g*}\{\bar{\ell}\}=0$), which is

$$g^*(x) = \frac{\varphi[M(X)] f(x,v)}{\ell} \quad (9)$$

If we get the $g*(x)$, the estimator (7) has zero variance, and we just need to produce one sample. But getting $g*(x)$ is difficult because ℓ is an unknown parameter. However, it is convenient to get a pdf in the family of densities of f(x). In CEM, g(x) is chosen from the family of densities $\{f(x,v'), v' \in M\}$ on X. Let $f(x)=f(x,v_0)$ and $g(x)=f(x,u)$, $u \neq v_0$. And the parameter u is called the reference parameter. Therefore, (8) can be written as

$$\min_u Var_u\{\varphi[M(X)] W(X, v_0, u)\} \quad (10)$$

Where, $W(X, v_0, u)$ is called the likelihood radio (LR), and it is

$$W(X, v_0, u) = \frac{f(X, v_0)}{g(X, u)}.$$

In GEM, the cross-entropy is used to get an optimal of u, says u^*. The cross-entropy[14] defines the distance between the two probability distributions $f(y)$ and $g(y)$ and can be written as

$$D(f,g) = \int f(x) \ln \frac{f(x)}{g(x)} dx$$

According to[13], u^* is

$$u^* = \arg\max_u \{E_{v_0}[\varphi[M(X)]] \ln f(X,u)\} \quad (11)$$

An estimator of u^* is

$$\hat{u}^* = \arg\max_u \left\{\frac{1}{N}\sum_{i=1}^{N}\varphi[M(X_i)] \ln f(X_i,u)\right\} \quad (12)$$

Where, $X_1, \cdots, X_N$ are random samples from $f(x, v_0)$. However, if $\{M(X)<\gamma\}$ is a rare event respect to the $f(x, v_0)$, we may generate lots of samples to get a small relative error or a narrow confidence interval. To overcome this difficulty, Rubinstein[17] introduces a updating process about γ and reference parameter u. An auxiliary sequence of (γ_t, u_t), $t \geqslant 0$, is used in the updating process. The γ_t is chosen to make the probability of $\{M(X)<\gamma_t\}$ is

not too small, says $\approx 10^{-2}$, at iteration t. The updating of the sequence is described as follows.

1) Adaptive updating of γ_t. For a fixed u_t, and $X_1, \cdots, X_N$, which are random samples from $f(x, u_t)$, it should satisfy

$$P_{u_t}(M(X) < \gamma_t \geqslant \eta)$$

To get an estimator of γ_t, we can order those samples from smallest to biggest respect to the value of $M(X)$: $M(N_1) \leqslant M(N_2) \leqslant \cdots \leqslant M(N_n)$. And, γ_t gets its value as

$$\gamma_t = M\left(\lceil (1-\eta)N \rceil\right) \tag{13}$$

Where, $\lceil (1-\eta)N \rceil$ is the smallest integer that not less than $(1-\eta)N$.

2) Adaptive updating of u_t. For a fixed u_t and γ_t, u_{t+1} can be derived from the following formula.

$$u_{t+1} = \arg\max_{u_{t+1}} \left\{ E_{u_t}\left[\varphi[M(X)]\right] \ln f(X, u_{t+1}) \right\} \tag{14}$$

The stochastic counterpart of (14) is

$$u_{t+1} = \arg\max_{u_{t+1}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \varphi[M(X_i)] \ln f(X_i, u_{t+1}) \right\} \tag{15}$$

where, $X_1, \cdots, X_N$ are random samples from f (x, u_t)。

In [17], Rubinstein shows the monotonicity of the sequence γ_t is an inherent part of CEM, and the target value γ, which is close to γ^*, can be reached with high probability in a finite number of iterations. Therefore, we can get an approximation solution of MCDS.

4 Backbone formulation algorithm based on CEM

4.1. Constructing a MIS

In this paper, a MIS formulation approach that mentioned in [12] is used. Before we construct the MIS, every node is assigned a rank. In this paper, the rank of node n is defined as rank(n) = (*degree*(n), n). The *degree*(n) refers to the number of neighbors of node n. As shown in (16), rank(n_1) is greater than rank(n_2) if and only if the degree (n_1) is greater than degree(n_2) or n_1 is greater n_2 when degree(n_1) equals degree(n_2).

$$rank(n_1) > rank(n_2) \Leftrightarrow \begin{cases} degree(n_1) > degree(n_1) \\ \text{or} \\ degree(n_1) = degree(n_1), n_1 > n_2 \end{cases} \tag{16}$$

The MIS constructing process follows two rules.

Rule1. All nodes in the network are initialized with white color.

Rule2. If a white node has a higher rank than all its white neighbors, it will be colored black and all of its neighbors will be colored gray.

Once a white node marks itself to black, it broadcasts a BLACK message to its neighbors. When a white node receives the BLACK message at the first time, it colors itself to gray, and broadcasts a GRAY message. If a white node has received GRAY messages from all of its neighbors of lower rank, it changes its color to black and broadcasts a BLACK message.

It is easy to prove that all nodes are marked black or gray after the process and these black nodes form a MIS (every MIS is also a DS). In this paper, the black node is called dominator and the gray node is called dominatee.

4.2 Getting the action sets

To get a CDS, we should find some dominatee to connect the MIS (those dominatee is called connector). In[12], it is shown that the short path between any two complementary subsets of MIS is either two or three hops path. Therefore, a CDS can be got if every get the paths to connect its two or three hops dominator neighbors. The method presented in[11] is utilized to make each dominator gets the shortest hops paths to its two or three hops dominator neighbors (note that, in this paper, all dominator are required to broadcast a TWO-HOP-DOMINATOR message[11]).

If we replace the connection path between any two dominators with an edge, to get a CDS with a small size, there should be no cycle in the CDS that we want to get. So, the topology of the CDS is a

tree. Every dominator (except the one chosen as the root) is required to choose another dominator as its father node. In this paper, a possible choice is called an action and an action set of a dominator that contains all of its possible choices. As shown in Fig. 1, one action set consists of two parts. The former consists of those dominators that may have a connection (we call those nodes as DNC). The second part contains connectors that included in the connection path to one DNC.

DNC	connectors
d1	c1
d2	(c2,c3)
d3	(c4,c5)
…	…

Fig. 1 An illustration of the action set

To connect a dominator that is two hops away, one connector should be added in the action set. A triple (a1, a2, a3) is used to represent a path from dominator a1 to its two hops dominator neighbor a3. If there are several paths between a1 and a3, the one with the connector of maximum rank is chose. For example, in Fig. 2, there are two paths (1, 4, 8) and (1, 5, 8) between node 1 and node 8. According to the principle, node 4 will be chose because it has a larger rank than node 5. This principle can decrease the size of the final CDS because the connector with a larger rank is more likely chose by other dominators is larger, for example, node 4 is also chose as a connector by node 10. To connect a dominator that is three hops away, two connectors should be added in the action set and a four-tuple (b1, b2, b3, b4) is used to represent a path. Similar with before, if there are multiple paths, the path that has a greater amount of degree is chose. For example, (1, 3, 6, 11) will be chose to be the path between node 1 and node 11.

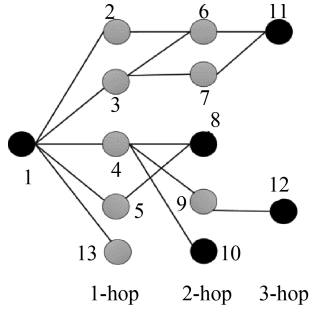


Fig. 2 An illustration of connector selecting

4.3 Choosing actions to connect MIS

All action sets of nodes are sent to the sink after every dominator gets its action set. And sink initializes the probability matrix Pr^t with the values in Pr^0. Pr^t is shown as follows

$$Pr^t = [\,Pr_1^t, Pr_2^t, \cdots Pr_m^t\,]^T \tag{17}$$

Where, t is the index of iteration, m is the number of dominator, $Pr_i^t = (\,P_{i,1}^t, P_{i,2}^t, \cdots P_{im}^t\,]^T$ the probability vector of the ith dominator. $P_{i,j}^t$ is the probability that the ith dominator chooses the jth dominator as its father node. And $P_{i,j}^t$ is initialized as

$$p_{i,j}^0 = \begin{cases} 0, & n_j \notin Z(i) \\ 1/|Z(i)|, & n_j \in Z(i) \end{cases} \tag{18}$$

Where, Z(i) is the set of dominators that in the action set of ith dominator. |Z(i)| is the number of dominators in Z(i). n_j is the jth dominator.

In the t iteration, some samples are got based on Pr^t and the parameters of CEM will be updated based on those samples. The updating of γ_t could be found in (13). The updating of Pr^t is shown as follows,

$$\mathbf{Pr}^{t+1} = \arg\max_{\mathbf{Pr}^{t+1}} \left\{ E_{\mathbf{Pr}^t} \left[\varphi[M(X)] \right] \ln f(X, \mathbf{Pr}^{t+1}) \right\} \tag{19}$$

In this paper, as shown in [15], Pr^t satisfies

$$\ln f(X, \mathbf{Pr}^{t+1}) = \sum_{i,j \in Z(i)} I_{\{X \in \tilde{X}_{(i,j)}\}} \ln p_{i,j}^{t+1} \tag{20}$$

Where $\tilde{X}_{(i,j)}$ is the set of backbones, in which n_i chooses n_j as its father node or n_j is a child of n_i. Note that $\sum_{nj \in z(i)} p_{i,j}^{t+1} = 1$ is satisfied for each dominator n_i. So we introduce the Lagrange multipliers λ_1, λ_2, …, λ_m to obtain

$$\mathbf{Pr}^{t+1} = \arg\max_{\mathbf{Pr}^{t+1}} \left\{ E_{\mathbf{Pr}^t} \left[\varphi[M(X)] \right] \ln f(X,u) + \sum_{i=1}^{m} \lambda_i \left(\sum_{n_j \in Z(i)} p_{i,j}^{t+1} - 1 \right) \right\} \tag{21}$$

When (21) get its solution, the following must be satisfied for all i= 1, 2, …m,

$$\frac{E_{\mathbf{Pr}^t}\left[\varphi[M(X)]\right]I_{\{X\in\tilde{X}_{(i,j)}\}}}{p_{i,j}^{t+1}}+\lambda_i=0,\quad \forall n_j\in Z(i) \tag{22}$$

Summing over | Z(i) | formulas that is shown in (22), we can get

$$-\lambda_i=E_{\mathbf{Pr}^t}\left[\varphi[M(X)]\right]I_{\{X\in\tilde{X}_{(i)}\}},$$

Where $\tilde{X}_{(i)}$ is the set of backbones that includes dominator n_i. Therefore, for all i= 1, 2, …m,

$$p_{i,j}^{t+1}=\frac{E_{\mathbf{Pr}^t}\left[\varphi[M(X)]\right]I_{\{X\in\tilde{X}_{(i,j)}\}}}{E_{\mathbf{Pr}^t}\left[\varphi[M(X)]\right]I_{\{X\in\tilde{X}_{(i)}\}}},\quad \forall n_j\in Z(i) \tag{23}$$

An estimator of (23) is

$$p_{i,j}^{'t+1}=\frac{\sum_{k=1}^{N}\left(I_{\{M(x)<\gamma_t\}}I_{\{X_k\in\tilde{X}_{(i,j)}\}}\right)}{\sum_{k=1}^{N}\left(I_{\{M(x)<\gamma_t\}}I_{\{X_k\in\tilde{X}_{(i)}\}}\right)},\quad \forall n_j\in Z(i) \tag{24}$$

where, $X_1,\cdots,X_N$ are random samples from Pr^t.

Similar to [15], a smoothed updating of Pr^t is used in our method. The smoothed updating is

$$p_{i,j}^{t+1}=(1-\alpha)p_{i,j}^{t}+\alpha p_{i,j}^{'t+1},\quad \forall n_j\in Z(i) \tag{25}$$

In summary, the backbone formulation algorithm based on cross-entropy method (BFA-CEM) is shown as the following table. Where, Max_i is the threshold of the number of iterations. And, after sink get the final CDS, it sends each dominator a Neighbor-Dominator message, in which the IDs of the father nodes and child node are included.

BFA-CEM (Backbone formulation algorithm based on cross-entropy method)

1. Getting a MIS.
2. Each dominator gets its action set and sends the set to sink.
3. Base on the CEM, sink gets a CDS.

   Initialize t=0 and Pr^0。

   While (t <Max_i)

   (1) Generate N samples based on Pr^t.

   (2) Update γ_t

   $$\gamma_t=M_{\lceil(1-\eta)N\rceil}.$$

   (3) Update Pr^{t+1}

   $$p_{i,j}^{'t+1}=\frac{\sum_{k=1}^{N}\left(I_{\{M(x)<\gamma_t\}}I_{\{X_k\in\tilde{X}_{(i,j)}\}}\right)}{\sum_{k=1}^{N}\left(I_{\{M(x)<\gamma_t\}}I_{\{X_k\in\tilde{X}_{(i)}\}}\right)},\quad \forall n_j\in Z(i)$$

   ;

   $$p_{i,j}^{t+1}=(1-\alpha)p_{i,j}^{t}+\alpha p_{i,j}^{'t+1},\quad \forall n_j\in Z(i).$$

   (4) t = t+1.

   End While

   Return the CDS that obtained in the last iteration and has the smallest size。

4. Sink sends each dominator a Neighbor-Dominator message。

5 Performance analysis

In this section, we show the time complexity, message complexity, and the approximation ratio of BFA-CEM. First, we define n is the number of nodes in the network, *opt* is the size of a MCDS of the network. According to Section 4.3, it is easy to get the

time complexity of BFA-CEM is $O(n^2 Max\_i)$.

Theorem 1. The message complexity of BFA-CEM is $O(n^2)$.

Proof. The transformation of packets consists of three phases: (i) a MIS is obtained; (ii) each dominator gets its action set; (iii) each dominator sends its action set to sink and gets a Neighbor-Dominator message from sink. At the first phase, each node broadcast a BLACK message or a GRAY message, the total number of messages is $O(n)$. At the second phase, each dominatee broadcasts two messages[11], the total number of messages is $O(n)$. At the third phase, the worst-case message complexity is $O(n^2)$. The worst case occurs when the size of MIS is n-1, and the number of hops of each transmission is n. To sum up, the message complexity of BFA-CEM is $O(n^2)$.

To get the approximation ratio of BFA-CEM, an important property of MIS is shown in Lemma 1.

Lemma 1. The size of any independent set in a UDG is at most $4opt+1$。

Proof. The proof can be found in[16]。

Theorem 2. The size of the CDS that got by BFA-CEM is at most $12opt+1$。

Proof. In the formulation process of CDS, each dominator (except the one that chose to be the root) will choose one node to be its father node from those its two and three hops dominator neighbors. According to the action set that shown in Section 4.2, the number of connector to complete a connection is at most 2. Based on the Lemma 1, the number of nodes in the CDS is at most $4opt+1+2\times 4opt = 12opt+1$.

6 Simulation results

In this section, several simulations are shown about the performance of our algorithm (the software of simulation is Matlab 2013b). We assume that nodes are uniformly distributed in a square area of 100 units by 100 units. The number of nodes in the network ranges from 100 to 500 (we represent this number as SN), and more than 50 connected graphs are random generated for each given number of node. We first get the influence of the values of η and α and compare the performance of BFA-CEM with Zone-based virtual backbone formation (ZVBF-MD)[3] and Torkestani's DAL-BF algorithm[8] in terms of the size of CDS, message overhead and the average node degree. As described in[8], the learning rate of DAL-BF is 0.2, the PCDS of DAL-BF is 0.9 and the maximum number of iterations of DAL-BF is 200. In the simulation of BFA-CEM, the Max_i is set to 20 and the number of samples in each iteration is set to 100. Similar to[15], if the best results of the last 5 iteration are the same, we thought the BFA-CEM is convergent and stop the BFA-CEM.

6.1 The influence of η

At the first, we show the influence of η. The simulation results are shown is Fig.3 and Fig. 4. In this simulation, α is set to 0.5 and SN is set to 200. The transformation range (we represent it as Tr) of node is set to 15, 20 and 25, respectively. The value of η changes from 0.02 to 0.12 with a step of 0.02. For ease of observation, as shown in Fig. 3, we compare the sizes of CDS that obtained in different η with the size of network when η is 0.02.We can get that a small value of η intend to get a less iterations. However, it also intends to get a large size when η is small. To make a trade-off between the size of the CDS and the number of iterations, based on those result, we let η is equal to 0.06 in the next simulations.

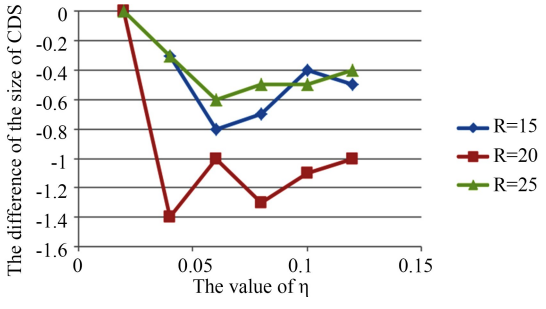


Fig. 3 The difference of the size of CDS in different η

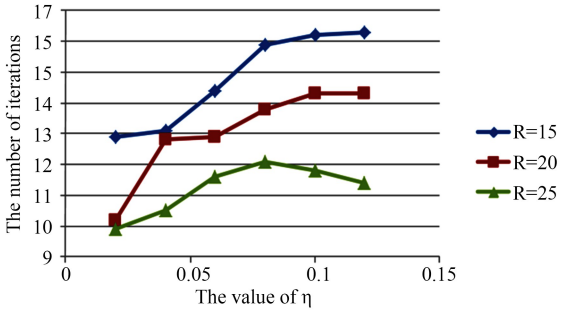


Fig. 4 The number of iterations in different η

6.2 The influence of α

In this section, we figure out how the value of α influences the size of network and the number of iterations. In this simulation, α is changed from 0.2 to 0.8 with a step of 0.2, and SN is set to 200. Tr is set to 15, 20 and 25, respectively. η is set to 0.06. Similar to before, for ease of observation, as shown in Fig. 5, we compare the sizes of CDS that obtained in different α with the size of network when α is 0.2. Fig. 6 shows the result about the number of iterations. We can get that the number of iterations decreases when α increase. And a larger α intend to get a CDS with a larger size. According to the formula (25), a larger α makes the convergence speed of BFA-CEM is faster. On the other hand, a faster convergence speed is more likely to lead the algorithm gets a local optimum.

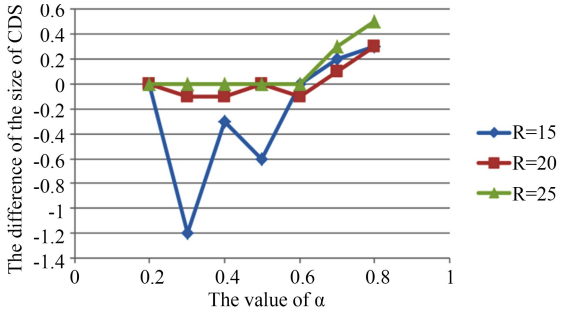


Fig. 5 The difference of the size of CDS in different α.

6.3 The size of network

In this section, we get the performance of BFA-CEM about the size of network. And we compare our algorithm with ZVBF-MD and DAL-BF. In those simulations, η is set to 0.06 and α is set to 0.3. Fig. 7 and Fig. 8 shows the results when SN is changed from 100 to 500. Tr that used in Fig.7 and Fig. 8 is set to 15 units and 30 units, respectively. On the other hand, Fig. 9 and Fig. 10 show the results about the size of network are shown when the value of Tr is changed from 15 to 40 units. And, in Fig. 9 and Fig. 10, SN is set to 100 and 200, respectively. According to those results, we can find that the size of CDS increases when SN increases. However, the increase rate is not obvious when SN is a large number (i.e. 400). This can be attached to the fact that a dominator covers more nodes when the network is dense. As shown in Fig. 9 and 10, the size of CDS decreases when Tr increases because the increase of Tr results in a dominator can cover more nodes. On the other hand, the number of nodes is fixed. Thus, fewer dominators are needed. We also notice that the performance of ZVBF-MD, BFA-PA and BFA-CEM are close when Tr is a large number.

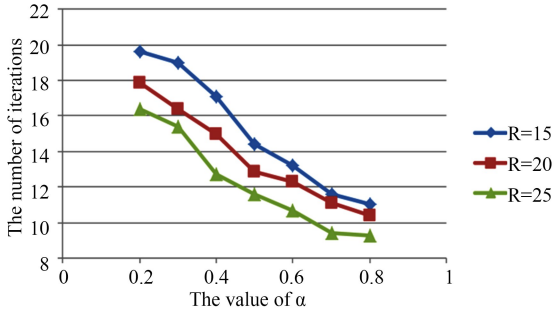


Fig. 6 The number of iterations in different α

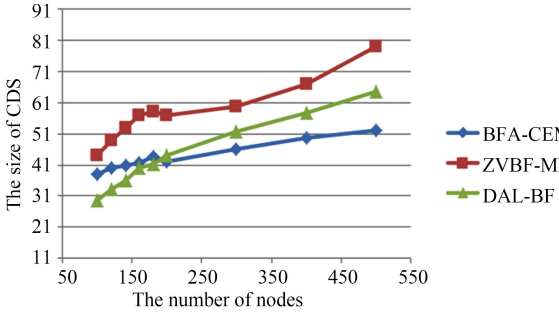


Fig. 7 The size of CDS when SN is changed and Tr equals 15 units

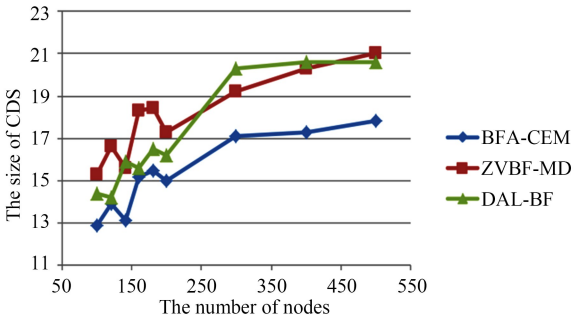


Fig. 8 The size of CDS when SN is changed and Tr equals 30 units

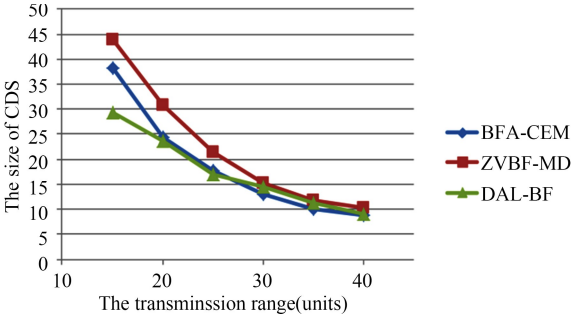


Fig. 9 The size of CDS when Tr is changed and SN equals 100

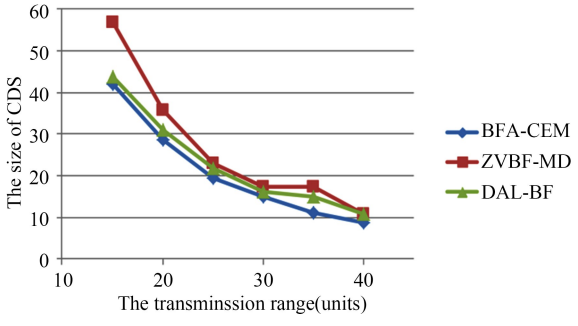


Fig. 10 The size of CDS when Tr is changed and SN equals 200

6.4 The message overhead

Message overhead is another important metric of a CDS formation algorithm. Fig. 11 and Fig. 12 show the simulation results about the message overhead of different algorithm when SN changes from 100 to 200. In the simulation, η is set to 0.06 and α is set to 0.3. Tr is set to 15 units and 30 units in Fig.11 and Fig. 12, respectively. Note that the message overhead of DAL-BF is not shown in those figures because its message overhead is too large than other two algorithms (i.e. the message overhead of DAL-BF is 400000 bytes, which is 60 times larger than the message overhead of ZVBF-MD, when SN is 100 and Tr is 15 units.). Based on those results, we can see that the message overhead is increases when SN or Tr increases. The message overhead of BFA-GEM is larger than the message overhead of ZVBF-MD, and it is more obvious when SN or Tr is large. In BFA-GEM, each dominator will get its action set and send the set to the sink. In ZVBF-MD, only the border dominators send its choice to the center node of same zone. Thus, the difference occurs. However, the message overhead of BFA-GE is not large and it is reasonable.

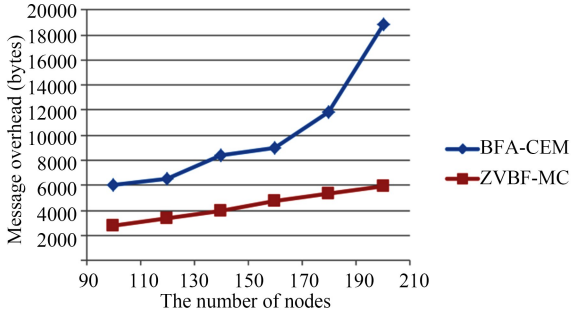


Fig. 11 The Average message overhead when SN is changed and Tr equals 15 units

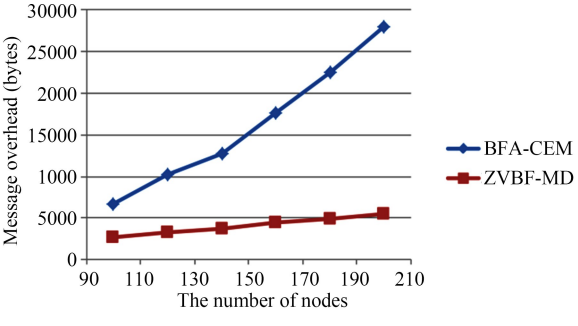


Fig. 12 The Average message overhead when SN is changed and Tr equals 30 units

6.5 The average node degree of backbone

In [11], Bo introduces the average node degree as a metric of the CDS. Generally, a small degree is

related to less interference of packet transmission. Fig. 13 and Fig.14 show the results about the average node degree of different algorithm. In this simulation, SN changes from 100 to 500, η is set to 0.06 and α is set to 0.3. In Figure 13, Tr is 15 units. And Tr is 30 units in Figure 14. We can find that the node degree of the CDS increases when the size of network increases and Tr increases because the network is more denser. However, the increase is not conspicuous any more when the number of node or the transmission range is large. According to those result, DLA-BF has a large average node degree. BFA-CEM has the smallest average node degree. Moreover, BFA-CEM and ZVBF-MD have a similar average node degree when Tr is 30.

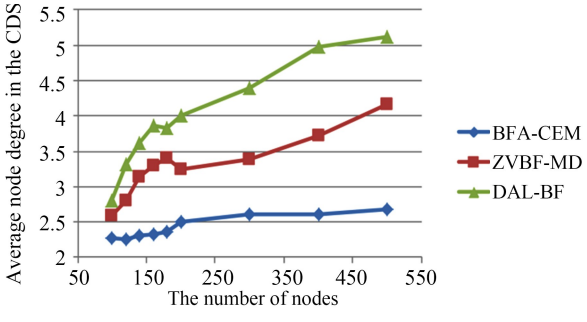


Fig. 13 Average node degree in the CDS when SN is changed and *Tr* equals 15 units

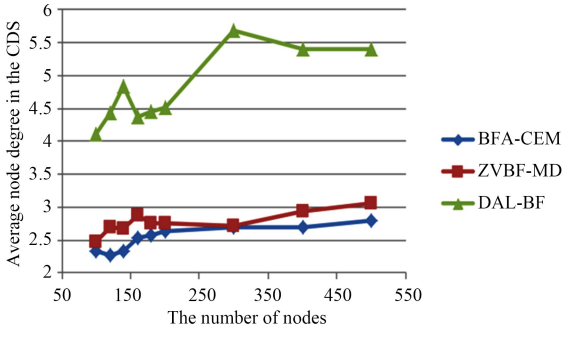


Fig. 14 Average node degree in the CDS when SN is changed and *Tr* equals 30 units

7. Conclusions

In this paper, based on the cross-entropy method, we propose BFA-CEM to get a backbone of WSN. In BFA-CEM, a MIS is got at first and dominators get their action sets. Finally, some connectors are choses to be connector to get a backbone. In BFA-CEM, CEM is utilized to increases the probability of getting an approximation solution of MCDS and decrease the number of samples by changing the threshold γ and probability matrix with time. We also analyze the performance of BFA-CEM. The simulation shows that our algorithm generates a smaller CDS than ZVBF-MD. And it has a smaller size of CDS than DLA-BF when the network is not sparse and has a similar performance to DLA-BF when the network is sparse. The message overhead of BFA-CEM is reasonable and its average node degree is small. This approach can be potentially used in designing efficient broadcasting strategy or working as a backup routing of WSN.

References

[1] AYDAY E, FEKRI F. A secure broadcasting scheme to provide availability, reliability andauthentication for wireless sensor networks[J]. Ad Hoc Networks, 2012, 10(7): 1278-1290.

[2] YU J, WANG N, WANG G, et al. Connected dominating sets in wireless ad hoc and sensor networks-A comprehensive survey[J]. Computer Communications, 2013, 36(2): 121-134.

[3] ALZOUBIK M, LI X Y, WANG Y, WAN P J, FRIEDER O. Geometric spanners for wireless ad hoc network[J]. IEEE Transactions on Parallel and Distributed Systems, 2003, 14 (4):408-421.

[4] LIY, THAI M T, WANG F, YI C W, WANG P J, DU D Z. On greedy construction of connected dominating sets in wireless networks[J]. Wireless Communications and Mobile Computing, 2005, 5 (8): 927-932.

[5] WU J, LIHL. On calculating connected dominating set for efficient routing in ad hoc wireless networks[C]. The Third ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (ACM DIALM 1999), August 1999, 7-14.

[6] DAIF, WU J.An extended localized algorithm for connected dominating set formation in ad hoc wireless networks[J]. IEEE Transactions on Parallel and Distribu-

ted Systems, 2004, 15 (10): 908-920.

[7] BUTENKOS, CHENG X, OLIVEIRA C, PARDALOS P M. A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks [C]. Recent Developments in Cooperative Control and Optimization, Kluwer Academic Publishers, 2004: 61-73.

[8] AKBARI TJ, MEYBODI M R. An intelligent backbone formation algorithm in wireless ad hoc networks based on distributed learning automata [J]. Computer Networks, 2010,54 :826-843.

[9] AKBARI T J. Energy-efficient backbone formation in wireless sensor networks [J]. Computers & Electrical Engineering, 2013, 39(6): 1800-1811.

[10] HE J, JI SJ, PAN Y, et al. Approximation algorithms for load - balanced virtual backbone construction in wireless sensor networks [J]. Theoretical Computer Science, 2013, 507: 2-16.

[11] HAN B. Zone - based virtual backbone formation in wireless ad hoc networks [J]. Ad Hoc Networks 7.1 (2009): 183-200.

[12] KALED M A, WAN PJ, FRIEDER O. Maximal independent set, weakly connected dominating set, and induced spanners for mobile ad hoc networks [C]. International Journal of Foundations of Computer Science, 2003 14(2):287-303.

[13] RUBINSTEIN R Y, MELAMED B. Modern simulation and modeling [M]. John Wiley & Sons, New York, 1998.

[14] KAPURJ N, KESAVANH K. Entropy optimization principles with applications [M]. Academic Press: 1992.

[15] DE BOER P T, KROESE D P, MANNOR S, et al. A tutorial on the cross-entropy method [J]. Annals of operations research, 2005, 134(1): 19-67.

[16] WAN P J, ALZOUBI K M, FRIEDER O. Distributed construction of connected dominating set in wireless ad hoc networks [C]. Infocom 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. IEEE, 2002, 3: 1597-1604.

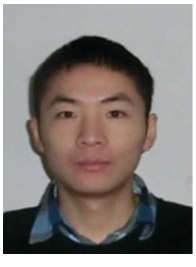
[17] RUBINSTEIN R. The cross-entropy method for combinatorial and continuous optimization [J]. Methodology and computing in applied probability, 1999, 1(2): 127-190.

Authors' Biographies

SHI Weiren received B. E. degree from Chongqing University in 1977, China. Now he is a professor and doctoral supervisor in Chongqing University. His main research directions include information control and intelligence system, wireless sensor network and embedded system.

Address: College of Automation, Chongqing University, Chongqing 400044, China.

Tel: +86-023-65106238; E-mail: wrs@cqu.edu.cn



JIANG Yisong received B. E. degree from Chongqing University in 2010, China. Now he is a PH. D. candidate in Chongqing University. His research interests include wireless sensor network and pervasive computing.

Address: College of Automation, Chongqing University, Chongqing 400044, China.

Tel: +086-13983282016; E-mail: jys12398@126.com

ZHAO Ying received B. Sc. and M. Sc. both from Chongqing University in 2011and 2014, respectively. Now, she is a Ph.D. candidate in Chongqing University. Her present research interest is complex system control theory and application, wireless sensor network.

Tel: +086-023-65106238; E-mail: zhaoying.enjoy@163.com