

# Constructing Reliable Virtual Backbones in Probabilistic Wireless Sensor Networks

Hai Liu, Yong Feng, Qian Qian and Bin Zhang

Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming, China

Most existing algorithms used for constructing virtual backbones are based on the ideal deterministic network model (DNM) in which any pair of nodes is either fully connected or completely disconnected. Different from DNM, the probabilistic network model (PNM), which presumes that there is a probability to connect and communicate between any pair of nodes, is more suitable to the practice in many real applications. In this paper, we propose a new algorithm to construct reliable virtual backbone in probabilistic wireless sensor networks. In the algorithm, we firstly introduce Effective Degree of Delivery Probability (EDDP) to indicate the reliable degree of nodes to transfer data successfully, and then exclude those nodes with zero EDDP from the candidate dominator set to construct a reliable connected dominating set (CDS). Moreover, each domineer selects the neighbor dominator with the maximum delivery probability to transfer data. Through simulations, we demonstrate that our proposed algorithm can remarkably prolong the network lifetime compared with existing typical algorithms.

*ACM CCS (2012) Classification:* Networks → Network algorithms → Control path algorithms → Network design and planning algorithms

Networks → Network types → Ad hoc networks → Mobile ad hoc networks

Networks → Network performance evaluation → Network simulations

*Keywords:* probabilistic wireless sensor networks, virtual backbone, reliable connected dominating set, delivery probability

## 1. Introduction

Wireless sensor networks (WSNs) are emerging as the desired environment for increasing numbers of military and civilian applications,

such as disaster control, environment and habitat monitoring, battlefield surveillance and health care applications [1], [2]. Sensor nodes usually use battery power, and thus their energy is very limited and not easily replenished. The energy scarcity of sensor nodes significantly limits the lifetime and performance of wireless sensor networks [3], [4]. Simultaneously, for the infrastructure-less and dynamic topology features, many routing protocols usually cause broadcasting storm in WSNs, which aggravates the energy consumption of nodes [5]. Sensor nodes deployed in the field can't be recharged to retain their energy resources; hence energy efficiency is a major issue in WSNs [6].

The virtual backbone (VB) can achieve better energy conservation and higher routing efficiency in wireless sensor networks. In a virtual backbone, nodes being in active state are dominators and mainly responsible for delivering data between dominators; nodes being in sleeping state are dominees and mainly responsible for delivering their own monitoring data to dominators. Nowadays, most existing algorithms for constructing virtual backbone are based on the ideal deterministic network model (DNM), where any pair of nodes in a network is either fully connected or completely disconnected. Although this model is widely used, it ignores the dependency between the environment condition (obstacles such as building, foliage etc.) and the strength of the emitted signal [7]. Beyond the "always connected" region, there is a transitional region where a pair of nodes is probabilistically connected. Such pairs

of nodes are not fully connected but reachable via the so called lossy links. As reported in [8], there are often much more lossy links than fully connected links in a WSN. Therefore, a more practical network model for WSNs is the probabilistic network model (PNM).

In PNM, there is a delivery probability associated with each link connecting a pair of nodes, which is used to indicate the possibility of successfully transmitting packages on the link. In order to guarantee the reliability of the virtual backbone, it is reasonable to select the links with higher delivery probability. In recent years, there have been some algorithms for constructing reliable virtual backbones in PNM, e.g., constructing a reliable minimum-size CDS (MCDS) [9], a load-balanced virtual backbone [10] and a reliable topology control in the whole network [11]. However, these algorithms do not take the reliability of dominators into account, and thus do not guarantee the reliability of the constructed virtual backbone.

In this paper, we propose a new algorithm to construct Reliable Virtual Backbone in PNM by constructing a reliable CDS (RVBP-CDS). Particularly, the main contributions of this paper are summarized as follows:

- (i) We propose a new algorithm to construct a reliable virtual backbone in probabilistic wireless sensor networks. The core idea of the algorithm is preferentially to improve the reliability of the links between dominators, and then improve the reliability of the links between dominators and dominatees.
- (ii) We introduce some new concepts in our algorithm, e.g., delivery probability rank to indicate the rank of delivery probability between a node and its neighbor nodes; effective degree of delivery probability to ensure the number of effective neighbor nodes.
- (iii) We propose a new method to construct a reliable CDS, in which the nodes which have the higher EDDP and delivery probability are preferentially selected as dominators.

The rest of this paper is organized as follows. In Section 2, we review some related literature on DNM and PNM, analyze its advantages and disadvantages. In Section 3, we introduce the net-

work model and delivery probability, formally define some concepts under PNM. The design of the RVBP-CDS algorithm is presented in Section 4. The simulation results are presented in Section 5 to validate our proposed algorithm. Finally, the paper is concluded in Section 6.

## 2. Related Work

The ideal deterministic network model (DNM) has been widely used in most existing algorithms for constructing virtual backbone and the connected dominating set is a good way to build the virtual backbone.

The authors in [12] proposed an algorithm for constructing the virtual backbone by constructing a connected dominating set. In this paper, dominator nodes are selected according to the number of neighbor nodes. Each node is initially colored white. Next, the node with the largest degree is colored black and all its neighbors are colored gray. This last step is repeated until there are no white nodes left in the graph. Each time, the gray node with the largest number of white neighbors is colored black and then all its white neighbors are colored gray. Node IDs can be used to break ties. Finally, all black nodes form a CDS. Since this CDS is also of minimal size, it is called MCDS. The MCDS can cause a single dominator with more connected dominatees to consume energy faster than other dominators.

In order to solve the unbalance-load problem, in [12], J. He *et al.* proposed an algorithm for constructing a load-balanced virtual backbone [13]. For achieving the purpose of load balancing about the whole network, there are two parts of work. The first part of the work is to construct a CDS with the minimum  $p$ -norm value in order to assure that the workload among each dominator is balanced. The second part of the work aims to load-balancedly allocate each dominatee to a dominator.

There are a lot of other algorithms to construct the virtual backbone, [14] proposed a grid partitioning algorithm for constructing the virtual backbone. The algorithm divides the whole area of the network into virtual grids. The virtual grid is defined such that, for any two adjacent grids, any node in one grid can directly commu-

nicate with any node in the other grid, and then choose a dominator in each virtual grid to form the virtual backbone. The work also proposed a state transition to change different node in active state to prolong the network lifetime. But the grid partitioning algorithm for constructing virtual backbone has a distinct disadvantage, which is that it can't guarantee the number of nodes in a grid, if the number of nodes in a grid is too little, it will reduce the whole network lifetime. Another disadvantage is that it can't guarantee the number of dominators to achieve the optimal.

In probabilistic network model, the algorithms for constructing the virtual backbone mainly include the following. A huge amount of approximation algorithms have been proposed to construct an MCDS-based VB, which is a well-known nondeterministic polynomial time (NP)-hard problem [15]. So we can only pray for an approximate solution. In [9], the author proposed a genetic algorithm for constructing a reliable MCDS in probabilistic wireless networks. Its reliability is above a preset application-specified threshold. The algorithm works mainly through using a fitness function to find a minimum-sized CDS whose reliability should be greater than or equal to a preset threshold. However, we can see that the reliability of CDS decreases while increasing number of dominators, i.e., increasing size of CDS, and the MCDS may cause single dominator which has more connected dominatees consumes its energy much faster than other dominators.

In [10], the authors proposed an algorithm for constructing a load-balanced virtual backbone in probabilistic wireless sensor networks via multi-objective genetic algorithm. On the basis of the original minimum connected dominating set, through the operation of mating and mutation in genetic algorithm to generate new connected dominating sets, the authors use dominating tree to select the load-balanced virtual backbone by comparing multi-objective fitness function. However, in the full text, the authors don't guarantee the reliability of the virtual backbone, which may reduce the network lifetime and increase the delay of network.

The authors in [11] firstly calculate the average delivery probability between all dominators, then calculate the average delivery probability

between each dominator and its dominatees, using the sum of the two probabilities as the gene in the multi-objective genetic algorithm. They use the binary tournament to select as the virtual backbone the best individual set, which is a CDS. However, the above method does not provide a good consideration of the link reliability of the constructed virtual backbone when used as the main data transmission channel. However, as dominators build the main data transmission channel in the network, the algorithm can't ensure that dominators have the highest reliability.

However, all of the abovementioned algorithms in probabilistic network model (PNM) have never considered the maximum reliability of dominators. Dominators are the main data transmission channel in the network. If we ensure the reliability of dominators, we will ensure the reliability of the whole network. In this paper, we preferentially consider the reliability of dominators, and then we improve the reliability between dominators and dominatees to the utmost extent.

### 3. Network Model and Problem Definition

#### 3.1. Network Model

The definition of network model in probabilistic network model (PNM) is given here. Under the PNM, we model a WSN as an un-directed graph  $G(V, E, \gamma(E))$ , where  $V$  is the set of  $n$  nodes, denoted by  $v_i$ , where  $0 \leq i \leq n$ ,  $i$  is called the node ID of  $v_i$  in the paper.  $E$  is the set of lossy links.  $\forall v_i, v_j \in V$ , there exists a link  $(v_i, v_j)$  in  $G$  if and only if:

- (i)  $v_i$  and  $v_j$  are in each other's transmission range and
- (ii)  $\gamma_{ij} > 0$ .

For each link  $(v_i, v_j) \in E$ ,  $\gamma_{ij}$  indicates the probability that node  $v_i$  can successfully and directly deliver a packet to node  $v_j$ , and  $\gamma(E) = \{\gamma_{ij} \mid (v_i, v_j) \in E, 0 < \gamma_{ij} \leq 1\}$ . We assume that the links are undirected (bidirectional), which means that two linked nodes are able to transmit and receive information from each other with the same  $\gamma_{ij}$  value [10]. Figure 1 presents an example of network model in this paper.

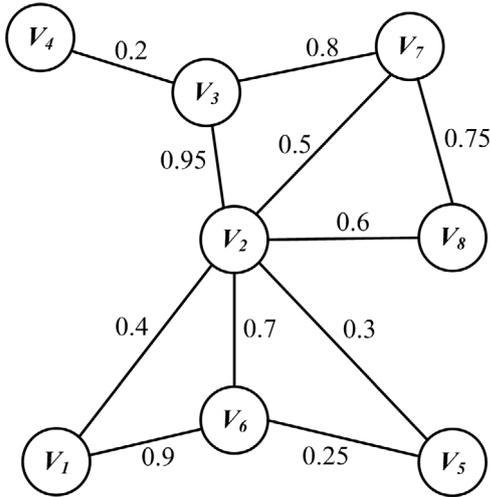


Figure 1. A wireless sensor network model in PNM.

### 3.2. Problem Definition

In this section, we first give the definition of delivery probability which has been defined in [16]. To solve the problem of improving the reliability of dominators, there are two aspects. The first aspect is to select the pairs of nodes which have the biggest delivery probability as dominators, but some edge nodes which have just one neighbor node can not satisfy the condition of dominator. The second aspect is to select the nodes which have the biggest number of neighbor nodes as dominators. However, although one node has the biggest number of neighbor nodes, all of the delivery probabilities between it and each neighbor node are relatively low. The case cannot improve the reliability of the constructed virtual backbone.

To solve the problem of the second aspect, we give the definition of delivery probability rank and delivery probability threshold to improve the delivery probability between the candidate dominator sets. To solve the problem of the first aspect, we give the definition of effective degree of delivery probability (EDDP) to indicate the reliable degree of nodes to transfer data successfully. So the problem becomes how to select the nodes which have the biggest delivery probability and EDDP. This will be described in Section 4.

#### 3.2.1. Delivery Probability

Although the ideal deterministic network model is widely used in most existing literature, it ignores the dependency on the condition of

the environment, such as the signal encoding, the strength of the emitted signal, the network payload, the signal-to-noise ratio, the ambient temperature and the transmitting distance. Beyond the "always connected" region, there is also a transitional region where any pair of nodes is probabilistically connected. Such pairs of nodes are not fully connected but are reachable via the so called lossy links. We define the lossy links as delivery probability.

We assume that the packet delivery probability is same as the packet reception probability. The packet reception probability is defined by  $\frac{1}{r} \sum_{i=1}^r P_i$  in literature [16], where  $r$  is the number

of tests. From the literature we know that the packet reception probability is inherently associated with the signal encoding and the sensor mote. With the change of the condition of the environment (such as the signal-to-noise ratio, the ambient temperature), the packet reception probability is also changed. When the condition of the environment and the signal encoding are confirmed, the packet reception probability varies exponentially with the distance between the nodes, as stated in formula (1). The following four formulas are the expression of the node packet reception probability in literature.

$$p(d) = \left( 1 - \frac{1}{2} \exp^{\frac{\gamma(d) \cdot 1}{2 \cdot 0.64}} \right)^{8f} \quad (1)$$

$$\gamma(d)_{dB} = P_{tdB} - PL(d)_{dB} - P_{ndB} \quad (2)$$

$$PL(d) = PL(d_0) + 10n \log_{10} \left( \frac{d}{d_0} \right) + X_{\sigma} \quad (3)$$

$$P_n = (F + 1)kT_0B \quad (4)$$

where  $f$  is the frame size,  $d$  is the transmitter-receiver distance,  $\gamma(d)_{dB}$  is the signal-to-noise ratio (SNR)  $\gamma$  at a distance  $d$  in formula (1). Given a transmitting power  $P_t$ ,  $PL(d)$  reflects the relation between the wireless channel and distance  $d$ ,  $P_n$  is the noise floor between the transmitter and receiver in formula (2). In formula (3),  $d_0$  is a reference distance,  $n$  is the path loss exponent, and  $X_{\sigma}$  is a zero-mean Gaussian Redundancy Version with standard deviation  $\sigma$ . Where  $F$

is the noise figure,  $k$  is the Boltzmann's constant,  $T_0$  is the ambient temperature and  $B$  is the equivalent bandwidth in formula (4).

### 3.2.2. Delivery Probability Rank

In probabilistic network model (PNM), there is a delivery probability between any two nodes. Delivery probability rank is a rank about delivery probability between the node and its neighbor nodes. The node with higher delivery probability has the higher rank. So each node will have a rank in the descending order about delivery probability. Here we use the 1-hop neighborhood to represent the delivery probability rank. Next we give the definition of 1-hop neighborhood.

**Definition 1.** 1-Hop Neighborhood ( $NP_1(v_i)$ ).  $\forall v_i \in V$ , the 1-Hop Neighborhood of node  $v_i$  is defined as

$$NP_1(v_i) = \{(v_i, \gamma_{ij}) \mid v_j \in V, \gamma_{ij} > 0\}$$

According to the definition 1 and Figure 1, we can get the 1-hop neighborhood of all nodes.

$$\begin{aligned} NP_1(v_1) &= \{(6, 0.9), (2, 0.4)\} \\ NP_1(v_2) &= \{(3, 0.95), (6, 0.7), (8, 0.6), \\ &\quad (7, 0.5), (1, 0.4), (5, 0.3)\} \\ NP_1(v_3) &= \{(2, 0.95), (7, 0.8), (4, 0.2)\} \\ NP_1(v_4) &= \{(3, 0.2)\} \\ NP_1(v_5) &= \{(2, 0.3), (6, 0.25)\} \\ NP_1(v_6) &= \{(1, 0.9), (2, 0.7), (5, 0.25), (7, 0.1)\} \\ NP_1(v_7) &= \{(3, 0.8), (8, 0.75), (2, 0.5), (6, 0.1)\} \\ NP_1(v_8) &= \{(7, 0.75), (2, 0.6)\} \end{aligned}$$

### 3.2.3. Delivery Probability Threshold

Having completed the delivery probability rank, on the basis of delivery probability rank, we first calculate the average delivery probability about each node and its neighbor nodes, and then we average all nodes' average delivery probability value, getting our delivery probability threshold.

$$\bar{\gamma}_i = \frac{\gamma_{i1} + \gamma_{i2} + \dots + \gamma_{ij}}{n} \quad (5)$$

$$\bar{\gamma} = \frac{\bar{\gamma}_1 + \bar{\gamma}_2 + \dots + \bar{\gamma}_n}{n} \quad (6)$$

where  $\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{ij} > 0$ ,  $n$  is the number of node  $v_i$ 's neighbor nodes in formula 5. The following is the average delivery probability of each node according to Figure 1.

$$\begin{aligned} \bar{\gamma}_1 &= 0.65 & \bar{\gamma}_2 &= 0.575 & \bar{\gamma}_3 &= 0.65 & \bar{\gamma}_4 &= 0.2 \\ \bar{\gamma}_5 &= 0.275 & \bar{\gamma}_6 &= 0.4875 & \bar{\gamma}_7 &= 0.5375 & \bar{\gamma}_8 &= 0.675 \end{aligned}$$

where  $\gamma_1, \gamma_2, \dots, \gamma_n > 0$ ,  $n$  is the number of all nodes in formula 6. Using the average delivery probability of each node and its neighbors, we can calculate the delivery probability threshold according to Formula 6, which amounts to 0.50625 for the network shown in Figure 1. We then take 0.6 to be the delivery probability threshold, which is an approximation slightly greater than 0.50625.

### 3.2.4. Effective Degree of Delivery Probability

Having calculated the delivery probability threshold, according to the delivery probability threshold, we remove the node whose delivery probability is lower than the delivery probability threshold in delivery probability rank. For each node  $i$ , its effective degree of delivery probability is defined as the number of the nodes satisfying the condition to be among the neighbors of node  $i$ , with the delivery probability of each of them not being lower than the delivery probability threshold. The following is the 1-hop neighborhood of all nodes about the EDDP, according to Figure 1.

$$\begin{aligned} NP_1(v_1) &= \{(6, 0.9)\} \\ NP_1(v_2) &= \{(3, 0.95), (6, 0.7), (8, 0.6)\} \\ NP_1(v_3) &= \{(2, 0.95), (7, 0.8)\} \\ NP_1(v_6) &= \{(1, 0.9), (2, 0.7)\} \\ NP_1(v_7) &= \{(3, 0.8), (8, 0.75)\} \\ NP_1(v_8) &= \{(7, 0.75), (2, 0.6)\} \end{aligned}$$

The pseudo-code of selecting the delivery probability threshold and the effective degree of delivery probability are shown in Algorithm 1.

*Algorithm 1.* Effective degree of delivery ratio.

- 
1. Initialization:  $\text{flag}(i) = 0$ ;  $\text{rank}[i][j] = 0$ ;
  2. **for** ( $i = 1$ ;  $i \leq N$ ;  $i++$ )
  3.   **for** ( $j \neq i$ ;  $j \leq N$ ;  $j++$ )
  4.     Descending sort  $\gamma_{ij}$ ,  $\text{rank}[i][j] = \gamma_{ij}$ ;
  5.     Calculate the  $\overline{\gamma}_{ij}$  of node  $i$ ;
  6.   **end for**
  7.   Calculate the  $\overline{\gamma}_{ij}$  of all nodes;
  8. **end for**
  9. Delete  $\text{rank}[i][j]$ , if  $\gamma_{ij} < \overline{\gamma}_{ij}$ ;
- 

## 4. RVBP-CDS Algorithm

### 4.1. Node Initialization

According to the effective degree of delivery probability defined in Section 3, we know that there are some nodes whose EDDP is zero, and the delivery probability between these nodes and their neighbor nodes is lower than the delivery probability threshold. The core idea of the algorithm is preferentially improving the reliability of dominators, that is to say, the delivery probability between the dominator is the biggest. So, before we select the dominator node, the first step is removing the node whose degree is zero and identifying some dominatees. We call this node initialization.

In this step, according to the EDDP, the node with zero degree is colored grey, other nodes are colored white. According to Figure 1 and the EDDP in Section 3, we know the EDDP about node 4 and node 5 is zero, so node 4 and node 5 are colored grey, other nodes are colored white in the initialization step.

### 4.2. Construction of a Reliable CDS

Connected dominating sets are one of the primary techniques used to build virtual backbones for wireless sensor and ad hoc networks. The concept of CDS can be briefly introduced as follows. A dominating set (DS) is a subset

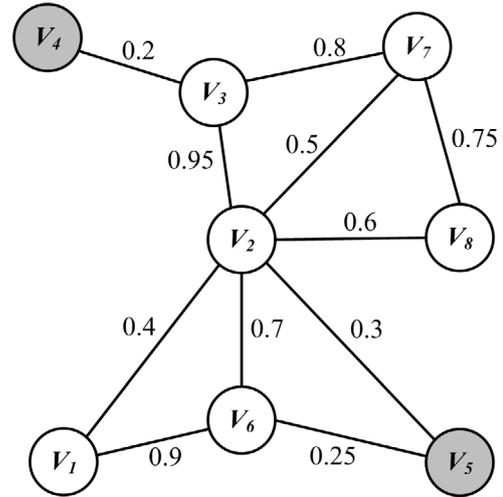


Figure 2. Node initialization.

of the vertices of a graph where every vertex in the graph is either in the subset or is adjacent to at least one vertex in the subset. A CDS is a connected DS. The nodes in CDS are called dominators, while other nodes are called dominatees [17].

In this paper, our method is to construct the CDS by growing it from a single node outward. The single node is also referred to as the first dominator node. We choose the node which has the biggest EDDP as the first dominator. If there is more than one node with the biggest EDDP, we choose the node which has the biggest delivery probability. The first dominator is colored black and all its uncolored neighbors are colored grey. According to Figure 2 and the EDDP in Section 3, we know the biggest EDDP of node 2 is 3, so node 2 is colored black, and its uncolored neighbors are colored grey.

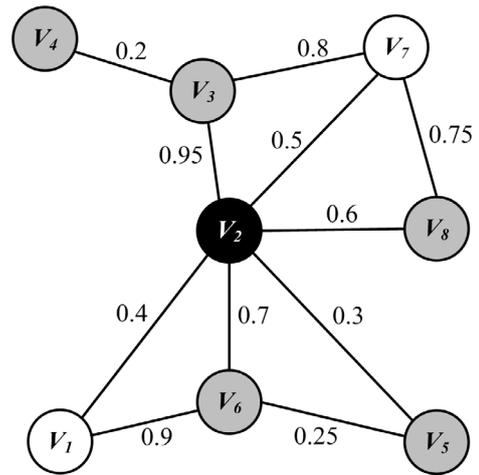
After the first black node is determined, we judge whether the next node is a black node by using the highest delivery probability rank in all black nodes. If there is a node with the biggest EDDP among all nodes directly connecting with the black nodes, but is not one of the black nodes, it is selected as the next dominator and colored black while all of its uncolored neighbors are colored grey. If such a node does not exist, then the delivery probability rank of the black nodes will be reduced by one level. The above procedure is repeated until the next dominator is selected out. According to Figure 3(a) and the EDDP calculated in Section 3, we know that the first black node is node 2, while

its delivery probability rank is 2. As node 3 has the highest link delivery probability between node 2 and its neighbors, which amounts to 0.95, additionally having the biggest EDDP except node 2, it is selected as the next dominator and colored black. Subsequently its uncolored neighbors are colored grey.

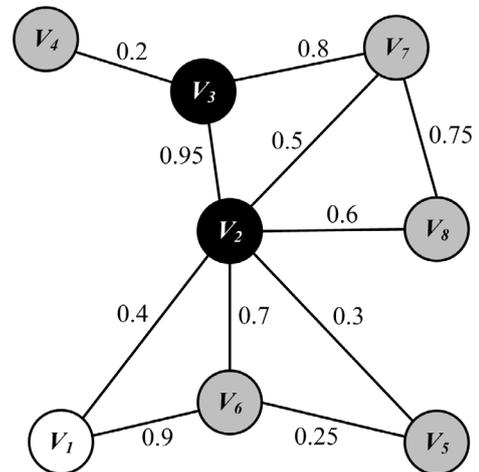
If one node and its neighbor nodes are colored grey or black, we also reduce the delivery probability rank in all black nodes' delivery probability rank. Repeat the above steps until all nodes are black or grey. As shown in Figure 3(b), the black nodes are node 2 and 3. Node 7 has the highest link delivery probability among such probabilities between any black node and its neighbor nodes, i.e. 0.8; node 7 has also the biggest EDDP except that of the black nodes. Hence, node 7 should have been selected as the next dominator but for the fact that it has already been colored, and is thus disqualified. Therefore, we try node 6 that has the second highest link delivery probability after node 7, namely 0.7. As node 6 is eligible, it is selected as the next dominator and colored black, after which its uncolored neighbors are colored grey. At this point, all nodes are colored black or grey, and the construction of reliable CDS accomplished.

### 4.3. Allocation of Dominatees

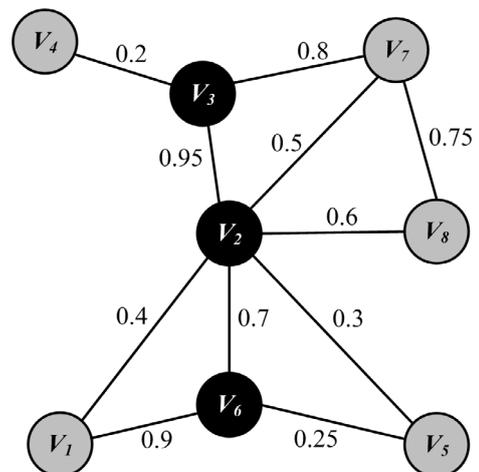
After we have successfully constructed the reliable CDS, the last step is the problem of the connection between dominatee and dominator. If a dominatee node has only one neighboring dominator node, then it delivers data to the dominator node; otherwise it delivers data to the dominator with the highest delivery probability among all its neighbor dominators. According to Figure 3(c) and the delivery probability rank table in Section 3, we know that node 4 just has one neighbor dominator, which is node 3, and node 8 has just one neighbor dominator, which is node 2, hence node 4 delivers data to node 3, while node 8 delivers data to node 2. Node 1 has more than one neighbor dominator, which are node 2 and node 6, node 5 has more than one neighbor dominator, which are node 2 and node 6, and node 7 has more than one neighbor dominator, which are node 2 and node 3. However, the delivery probability between node 1 and node 6 is 0.9, which is the highest delivery



(a) Choose the first dominator.



(b) Node traverse.



(c) Breaking the tie.

Figure 3. Construction of a reliable CDS.

probability in node 1; the delivery probability between node 5 and node 2 is 0.3; which is the highest delivery probability in node 5, and the delivery probability between node 7 and node 3 is 0.8, which again is the highest delivery probability in node 7. Hence, node 1 delivers data to node 6, node 5 delivers data to node 2, and node 7 delivers data to node 3.

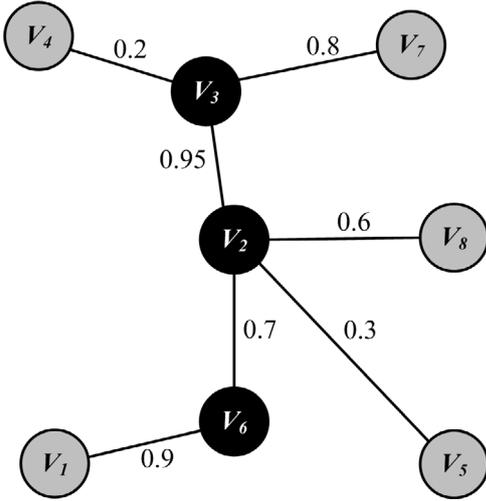


Figure 4. Allocation of dominattees.

The pseudocode of constructing the RVBP-CDS is shown in Algorithm 2.

Algorithm 2. RVBP-CDS.

1. **for** ( $i = 1$ ;  $i \leq N$ ;  $i++$ )
2.   **if**  $\text{rank}[i][j] = 0$ ,  $\text{flag}(i) = 2$ ;
3.   **else**  $\text{degree}[i] = j$ ;
4. **end for**
5. **if**  $\text{degree}[i]$  is the MAX;
6.    $\text{flag}(i) = 1$ , all  $\text{flag}(j) = 2$ ;
7. **end if**
8. **if** the MAX  $\text{degree}[i] > 1$ , select the MAX  $\gamma_{ij}$ , **do** 6;
9. **for** all nodes with  $\text{flag} = 1$ , select the MAX  $\gamma_{ij}$ ;
10. **if**  $\text{degree}[j]$  is the MAX except  $\text{degree}[i]$ ,
11. **do** 6,
12. **end if; end for;**
13. repeat 4,5 and 6, until all the node's  $\text{flag} = 1$  or 2;
14. **for all** the nodes with  $\text{flag} = 2$ ,
15. if they have one neighborhood with  $\text{flag} = 1$ , select it;
16. if they have more than one neighborhood with  $\text{flag} = 1$ , select the highest  $\gamma_{ij}$ ;
17. **end if; end for;**

## 5. Simulation Results

We compare our RVBP-CDS algorithm with the LBVBP-MOGA algorithm proposed in [10] and the RMCDS-GA algorithm proposed in [9] through the simulation experiment. We compare them in six main aspects: the network lifetime; the network delay; the packet delivery rate, the number of packet retransmissions, the average residual energy and the average dissipated energy. We compare the three algorithms in terms of network lifetime, which is defined as the time duration until the first dominator runs out of energy.

We build our own simulator where all the nodes have the same transmission range and all the nodes are deployed uniformly and randomly in a square area. For each specific setting, 100 instances are generated. Moreover, we assume that the node at the top end of the network is the the sink node, and data packets of all other nodes will be delivered to it. Furthermore, every sensor node produces a packet with size one during each communication interval. During the data transmission, we assume that the 1-hop transmission time is 0.1 s, if the data transmission fails, and then continue retransmission of this package after 1 s. The simulated energy consumption model is that every node has the same initial 1000 units of energy. Receiving and transmitting a packet both consume 1 unit of energy. We assume that the communication interval changes from 3 s to 30 s and increases 3 s every time. The simulation parameters are listed in Table 1.

Table 1. Simulation parameters.

Parameter	Value
Node initial energy	1000 (unit)
Number of sensor	8
Node transmit energy	1 (unit/packet)
Node receive energy	1 (unit/packet)
1-hop transmission time	0.1 (s)
Retransmission time interval	1 (s)
Communication interval	3 ~ 30 (s)
Packet numbers	1 (packet/interval)

### 5.1. Network Lifetime

Figure 5 shows the network lifetime of three algorithms (RMCDS-GA, LBVBP-MOGA, RVBP-CDS) under the change of communication interval. As we can see from the figure, the LBVBP-MOGA algorithm is better than the RMCDS-GA algorithm and our RVBP-CDS algorithm when the communication interval is less than 9 s. After more than 9 s, the RMCDS-GA algorithm and our RVBP-CDS algorithm are superior to the LBVBP-MOGA algorithm step by step. The cause for such behavior is that both algorithms – the RMCDS-GA and our RVBP-CDS – ensure the highest delivery probability between dominators, hence a packet generated by the source node can be delivered to the sink node in a short time. Thus the sink node shows apparent energy consumption, with a reduction of network lifetime. However, LBVBP-MOGA has a low reliability in the nodes, i.e. a large number of data packets are blocked in the nodes with low delivery probability when the communication interval is relatively low; consequently the sink node has lower energy consumption, hence the respective network lifetime is longer than the one in our algorithm. In the RMCDS-GA algorithm, the reliable minimum connected dominating set was used to construct the reliable virtual backbone, hence the respective network lifetime is longer than that of LBVBP-MOGA. But RMCDS-GA may create a situation in which the more dominatees a dominator owns, the faster the energy of the dominator is consumed. The uneven energy

consumption of RMCDS-GA leads to a network lifetime which is shorter than that of our RVBP-CDS.

### 5.2. Network Delay

The network delay is defined as the average number which is computed by dividing the total delay by the number of all generated packets. Here, the total delay is the sum of the delays for transmission or retransmission of each packet. Figure 6 shows the network delay of three algorithms (RMCDS-GA, LBVBP-MOGA, RVBP-CDS) under the change of communication interval. As we can see from the figure, the network delay of three algorithms reduces gradually. Finally, with the increase of communication interval, the network delay of LBVBP-MOGA algorithm is stable for about 13 s, and the RMCDS-GA algorithm and our RVBP-CDS algorithm are stable for about 1 s. Analyzing the reason, the RMCDS-GA algorithm and our RVBP-CDS algorithm ensure the delivery probability between the nodes. Especially our RVBP-CDS algorithm, we preferentially consider the reliability of dominators, and then we improve the reliability between dominators and dominatees to the utmost extent. In this way, we can reduce the amount of data retransmission times, and then reduce the delay of the whole network. But the LBVBP-MOGA algorithm has a low reliability in nodes, which causes a lot of packet retransmissions, and then increase the delay of the whole network.

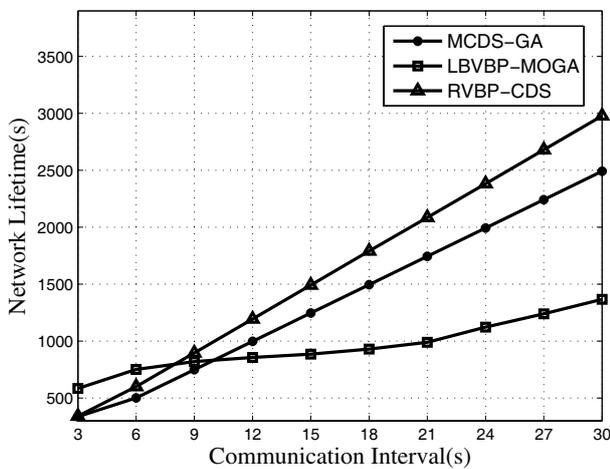


Figure 5. Network lifetime.

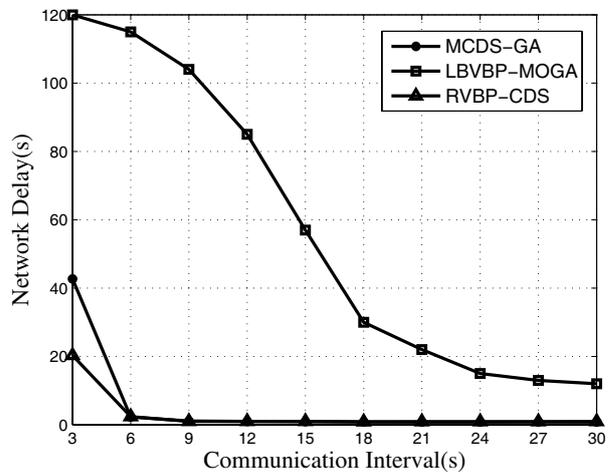


Figure 6. Network day.

### 5.3. Packet Delivery Rate

The packet delivery rate is calculated by dividing the number of all generated packets by the number of all packets which arrived at the sink node. Figure 7 shows the packet delivery rate of three algorithms (RMCDS-GA, LBVBP-MOGA, RVBP-CDS) under the change of communication interval. As we can see from the figure, the packet delivery rate of three algorithms increases gradually. Finally, with the increase of the communication interval, the packet delivery rate of LBVBP-MOGA stabilizes at around 0.85, while the ones of both RMCDS-GA and our RVBP-CDS at around 1.0. Analyzing the reason, the RMCDS-GA algorithm and our RVBP-CDS algorithm ensure the delivery probability between the nodes, so packets can be transmitted to the sink node at a rapid rate, and the energy consumption in each node is also more balanced, so they have a higher packet delivery rate than the LBVBP-MOGA algorithm. But the LBVBP-MOGA algorithm has a low reliability between the nodes, which causes a lot of packet retransmissions, and this will run out of the dominator energy with a lower probability, last it causes a fracture of the link and a large number of data packets can't be transmitted to the sink node.

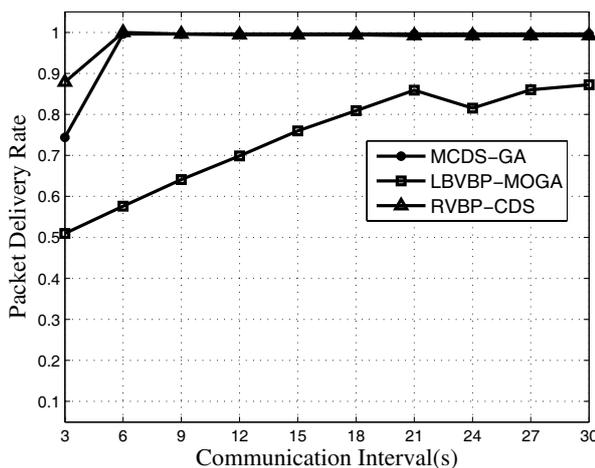


Figure 7. Packet delivery rate.

### 5.4. Retransmission Times

Retransmission times are calculated by dividing total retransmission times by the number of all generated packets. Here, the total retransmis-

sion time is the sum of retransmission times for each packet. Figure 8 shows the retransmission times of three algorithms (RMCDS-GA, LBVBP-MOGA, RVBP-CDS) under the change of communication interval. As we can see from the figure, the retransmission times of three algorithms increase gradually. Finally, with the increase of communication interval, the retransmission times of original LBVBP-MOGA algorithm is stability about 5, and the RMCDS-GA algorithm and our RVBP-CDS algorithm is stability about 1. Analyzing the reason, the RMCDS-GA algorithm and our RVBP-CDS algorithm ensure the delivery probability between the nodes, which greatly reduces the loss of data, thereby reducing the retransmission times. But the LBVBP-MOGA algorithm has a low reliability between the nodes, which increases the loss of data, and large amount of data packets need to be retransmitted.

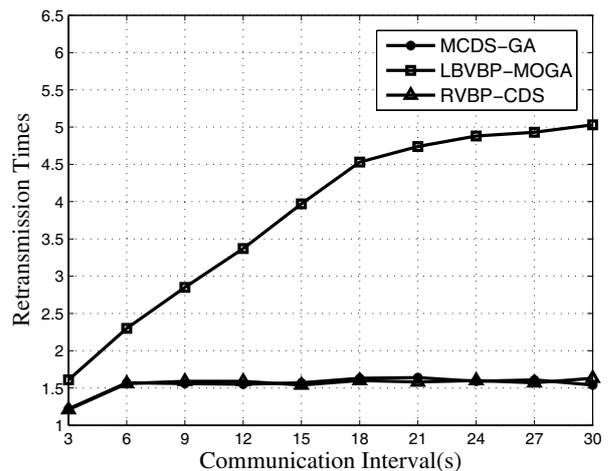


Figure 8. Retransmission times.

### 5.5. Average Residual Energy and Average Dissipated Energy

Average residual energy is calculated so that the total residual energy of all nodes is divided by the number of all nodes and average dissipated energy is calculated so that the total dissipated energy of all nodes is divided by the number of all nodes when the first dominator runs out of energy. Figure 9 and Figure 10 show the average residual energy and average dissipated energy of three algorithms (RMCDS-GA, LBVBP-MOGA, RVBP-CDS) under the change of

communication interval. As we can see from the figure, in both, the RMCDS-GA algorithm and our RVBP-CDS algorithm, average residual energy and average dissipated energy are stable at a fixed value under the change of communication interval, and the average residual energy of LBVBP-MOGA algorithm increases gradually. Finally, it is stability about 700, and the average dissipated energy of LBVBP-MOGA algorithm increases gradually, finally it is stability about 300 under the change of communication interval. Analyzing the reason, the RMCDS-GA algorithm and our RVBP-CDS algorithm ensure the delivery probability between the nodes, reduce the retransmission times, so they have a low dissipated energy and high residual energy.

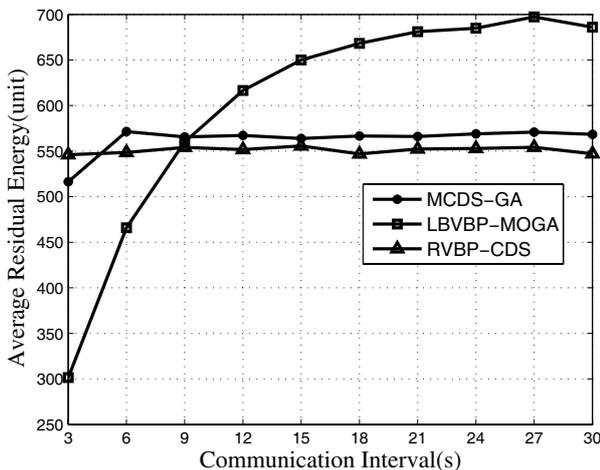


Figure 9. Average residual energy.

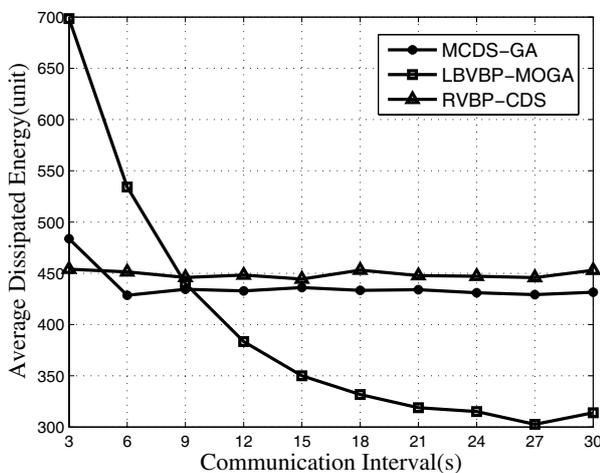


Figure 10. Average dissipated energy.

## 6. Conclusion

In this paper, we address the topic of constructing a reliable virtual backbone in probabilistic wireless sensor networks. On the basis of removing the nodes with zero EDDP, we construct a reliable CDS through growing it from the node with the highest EDDP outward, step by step. And in the dominatee allocation, each dominatee selects the neighbor dominator node with the highest delivery probability to deliver data. The simulation results demonstrate that using a reliable CDS can significantly prolong the network lifetime. Specifically, our proposed algorithm prolongs the network lifetime by 118% to the maximum compared with the LBVBP-MOGA algorithm and 20% to the maximum, compared with the RMCDS-GA algorithm.

## Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grants No. 61662042, 61262081, and 61462056; the Yunnan Provincial Key Project of Applied Basic Research Plan under Grants No. 2014FA028; the Fundamental Research Funds for the Central Universities under Grants No. ZYGX2012J083.

## References

- [1] S. Hadim and N. Mohamed, "Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks", *IEEE Distributed Systems Online*, vol. 7, no. 3, pp. 1–1, 2006. <http://dx.doi.org/10.1109/MDSO.2006.19>
- [2] F. Ishmanov *et al.*, "Energy Consumption Balancing (ECB) Issues and Mechanisms in Wireless Sensor Networks (WSNs): a Comprehensive Overview", *Eur. Trans. Telecomms*, vol. 22, pp. 151–167, Feb. 2011. <http://dx.doi.org/10.1002/ett.1466>
- [3] C. Schurgers and M. B. Srivastava, "Energy Efficient Routing in Wireless Sensor Networks", in *2001 Military Communications Conference for Network-Centric Operations: Creating the Information Force*, McLean, 2001, pp. 357–361. <http://dx.doi.org/10.1109/MILCOM.2001.985819>
- [4] X. Liang *et al.*, "Energy Efficient Modulation Design for Wireless Sensor Networks", in *2007 IEEE Pacific Rim Conference on Communica-*

- tions, *Computers and Signal Processing*, Victoria, 2007, pp. 98–101.  
<http://dx.doi.org/10.1109/PACRIM.2007.4313186>
- [5] Y. C. Tseng *et al.*, "The Broadcast Storm Problem in a Mobile Ad Hoc Network", *Wireless Networks*, vol. 8, no. 2, pp. 153–167, 2002.  
<http://dx.doi.org/10.1023/A:1013763825347>
- [6] S. B. H. Shah *et al.*, "OPEN: Optimized Path Planning Algorithm with Energy Efficiency and Extending Network – Lifetime in WSN", *Journal of Computing and Information Technology*, vol. 25, no. 1, pp. 1–14, 2017.  
<http://dx.doi.org/10.20532/cit.2017.1003259>
- [7] E. A. Khalil and S. Ozdemir, "Energy Aware Evolutionary Routing Protocol with Probabilistic Sensing Model and Wake-up Scheduling", in *2013 IEEE Globecom Workshops (GC Wkshps)*, Atlanta, 2013, pp. 873–878.  
<http://dx.doi.org/10.1109/GLOCOMW.2013.6825099>
- [8] A. Cerpa *et al.*, "Statistical Model of Lossy Links in Wireless Sensor Networks", in *Fourth International Symposium on Information Processing in Sensor Networks*, Boise, 2005, pp. 81–88.  
<http://dx.doi.org/10.1109/IPSIN.2005.1440900>
- [9] J. He *et al.*, "A Genetic Algorithm for Constructing a Reliable MCDS in Probabilistic Wireless Networks", in *2011 International Conference on Wireless Algorithms, Systems and Applications*, 2011, pp. 96–107.  
[http://dx.doi.org/10.1007/978-3-642-23490-3\\_9](http://dx.doi.org/10.1007/978-3-642-23490-3_9)
- [10] J. He *et al.*, "A Multi-Objective Genetic Algorithm for constructing Load-Balanced Virtual Backbones in Probabilistic Wireless Sensor Networks", in *2013 IEEE Global Communications (GLOBECOM) Conference*, Atlanta, 2013, pp. 261–266.  
<http://dx.doi.org/10.1109/GLOCOM.2013.6831081>
- [11] E. A. Khalil and S. Ozdemir, "CDS Based Reliable Topology Control in WSNs", in *2015 International Symposium on Networks, Computers and Communications (ISNCC)*, Hammamet, 2015, pp. 1–5.  
<http://dx.doi.org/10.1109/ISNCC.2015.7238569>
- [12] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets", *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.  
<http://dx.doi.org/10.1007/PL00009201>
- [13] J. He *et al.*, "Constructing a Load-Balanced Virtual Backbone in Wireless Sensor Networks", in *2012 International Conference on : Computing, Networking and Communications (ICNC)*, Maui, 2012, pp. 959–963.  
<http://dx.doi.org/10.1109/ICNC.2012.6167568>
- [14] Y. Xu *et al.*, "Geography-Informed Energyconservation for Ad Hoc Networks", in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, 2001, pp. 70–84.  
<http://dx.doi.org/10.1145/381677.381685>
- [15] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", 1th ed. New York: W. H. Freeman & Co., 1979.
- [16] M. Zuniga and B. Krishnamachari, "Analyzing the Transitional Region in Low Power Wireless links", in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, 2004, pp. 517–526.  
<http://dx.doi.org/10.1109/SAHCN.2004.1381954>
- [17] A. Nayak and I. Stojmenovic, "Energy-Efficient Backbones and Broadcasting in Sensor and Actuator Networks", in *Wireless Sensor and Actuator Networks Algorithms and Protocols for Scalable Coordination and Data Communication*, 1th ed. New Jersey: John Wiley & Sons, 2010, ch. 2, pp. 33–74.

Received: July 2016

Revised: May 2017

Accepted: June 2017

Contact addresses:

Hai Liu

Yunnan Key Laboratory of Computer Technology Applications  
 Kunming University of Science and Technology  
 Kunming, Yunnan 650500  
 China  
 e-mail: liuhai2046@sina.com

Yong Feng

Yunnan Key Laboratory of Computer Technology Applications  
 Kunming University of Science and Technology  
 Kunming, Yunnan 650500  
 China  
 e-mail: fybraver@163.com

Qian Qian

Yunnan Key Laboratory of Computer Technology Applications  
 Kunming University of Science and Technology  
 Kunming, Yunnan 650500  
 China  
 e-mail: qianqian\_yn@126.com

Bin Zhang

Yunnan Key Laboratory of Computer Technology Applications  
 Kunming University of Science and Technology  
 Kunming, Yunnan 650500  
 China  
 e-mail: zhangbin891130@126.com

---

HAI LIU is a Master graduate student of professional computer technology at Yunnan Key Laboratory of Computer Technology Applications, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, China.

---

---

YONG FENG received his PhD degree in Information and Communication Engineering from University of Electronic Science and Technology of China in 2011. He is currently an Associate Professor at Yunnan Key Laboratory of Computer Technology Applications, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, China. His research interests include Wireless Sensor Networks, Vehicular Ad Hoc Networks, and Internet of Things.

---

---

QIAN QIAN received his PhD degree in Kochi University of Technology, Japan, in 2011. He is currently an Associate Professor at Yunnan Key Laboratory of Computer Technology Applications, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, China. His research interests include Artificial Intelligence, and Visual Perception.

---

---

BIN ZHANG is a Master graduate student of professional computer technology at Yunnan Key Laboratory of Computer Technology Applications, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Yunnan, China.

---