

## Research Article

# A Fuzzy-Decision Based Approach for Composite Event Detection in Wireless Sensor Networks

Shukui Zhang,<sup>1,2</sup> Hao Chen,<sup>1</sup> Qiaoming Zhu,<sup>1</sup> and Juncheng Jia<sup>1</sup>

<sup>1</sup> School of Computer Science and Technology, Soochow University, Suzhou 215006, China

<sup>2</sup> State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Correspondence should be addressed to Shukui Zhang; zhangsk2000@163.com

Received 8 May 2014; Accepted 10 June 2014; Published 21 July 2014

Academic Editor: Hongli Xu

Copyright © 2014 Shukui Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The event detection is one of the fundamental researches in wireless sensor networks (WSNs). Due to the consideration of various properties that reflect events status, the Composite event is more consistent with the objective world. Thus, the research of the Composite event becomes more realistic. In this paper, we analyze the characteristics of the Composite event; then we propose a criterion to determine the area of the Composite event and put forward a dominating set based network topology construction algorithm under random deployment. For the unreliability of partial data in detection process and fuzziness of the event definitions in nature, we propose a cluster-based two-dimensional  $\tau$ -GAS algorithm and fuzzy-decision based composite event decision mechanism. In the case that the sensory data of most nodes are normal, the two-dimensional  $\tau$ -GAS algorithm can filter the fault node data effectively and reduce the influence of erroneous data on the event determination. The Composite event judgment mechanism which is based on fuzzy-decision holds the superiority of the fuzzy-logic based algorithm; moreover, it does not need the support of a huge rule base and its computational complexity is small. Compared to COLLECT algorithm and CDS algorithm, this algorithm improves the detection accuracy and reduces the traffic.

## 1. Introduction

In the event-driven wireless sensor networks (EWSNs) [1], a large number of sensor nodes deployed in monitoring area detect interested physical phenomena (such as forest fire prevention) in the environment. Event is the state changes in the real world [2]. Most previous studies of event detection just consider simple events. The literature [3] proposes distributed Bayesian algorithm for the area detection of the fault-tolerant events; this algorithm determines whether an event has occurred in its self-position through exchanging information with surrounding nodes. The literature [4] shows three ways to determine whether the nodes are at the boundary of the event; these three methods are based on statistics, image processing algorithm, and classification algorithm, respectively. The position of the node and judgment will be sent to the base station, and then the base station judges the area of the event. The literature [5] presents fault-tolerant events area detection mechanism by combining with space-time correlation and threshold. The literature [6] analyzes

the simple event detection delay; only a kind of sensor is needed in the network in the judgment of the simple event. However, most events need to detect various properties of the environment, if sensors that are needed to detect the Composite event can be integrated on a node, and then Composite event can be regarded as a simple event [7].

In recent years, heterogeneous wireless sensor network [8] (WHSN) consisted of different types of nodes has appeared; different types of nodes in the network integrate into different sensors. For example, there may be three kinds of nodes in the wireless sensor network (WSN) to detect fire; temperature, smoke concentration, and light intensity sensors are integrated into each node separately.

Wireless heterogeneous sensor network (WHSN) is suitable for Composite event detection [9], and there are two ways to deploy the nodes: nonrandom deployment and random deployment. For these two kinds of deployment, there are many literatures that have discussed the distributed algorithms. In the literature [10, 11], the monitoring area is divided into multiple virtual grids with the same size, and

each grid has a gateway node. The activated nodes provide the gateway node with environmental information, and then the gateway node judges whether the Composite event has occurred. Such algorithms assume that the sensing range of the node can cover the whole grid, and there are sufficient nodes in the grids to determine whether the Composite event occurs. The literature [12] studies the coverage of Composite events on the basis of dividing the grid. Whether Composite event occurs can be determined through exchanging information with neighboring under random deployment. According to the literature [13], COLLECT algorithm is proposed for Composite event detection and tracking problems. Since COLLECT algorithm requires that each node be active, therefore the traffic of this algorithm will be heavy when the event occurs. The literature [14] shows a method which is based on connected dominating set (CDS) for Composite events detection, but the accuracy is not high. Compared with COLLECT algorithm, CDS algorithm detects the Composite events by the sleep scheduling mechanism, and it reduces the network energy consumption effectively. In addition, the algorithm only needs active nodes to know their own locations; thus it reduces the position overhead.

A reliable routing mechanism is needed to ensure that the event information can be transmitted to the base station successfully after the events are detected. Reliable routing mechanism often establishes multiple paths between the source node and the base station, and event information can be sent to the base station as long as at least one path is valid. The literatures [15, 16] establish multiple disjoint paths between the source node and the base station to ensure the reliability of data transmission. The literature [17] proposes a SSR protocol, which requires data packets to consider the situation of surrounding nodes and choose the next-hop node properly in every transmission. Cut point will appear in the network when the node fails. When the cut point fails, the event information could not be transmitted to the base station, if all the paths between the source node and the base station pass the cut point. Therefore, it is necessary to fix the cut points. The literature [18] summarizes the existing cut point predicting algorithm and analyzes its performance. The literature [19] discusses how to repair the cut points in the case that there are mobile nodes in the network. When the link that reaches the next hop in the routing table fails, flooding [20] can be used to set a new route to forward the event information.

The above literatures have researched the network topology and routing. They all use threshold-based determination mechanisms; that is, when a node senses that a property's value is more than a given threshold, it considers that event related to the property occurs. In the literature [4], each node uses this strategy to determine whether the simple event occurs, then exchanges information with neighbor nodes, and finally determines their own verdict to ensure the boundaries of simple events. In the literatures [2, 10], the Composite event detection is achieved by the following process: first of all, whether simple events corresponding to different sensors reach a given threshold, Boolean calculation is operated on the result of these events to get the result of the determined events. However, it is not reasonable to determine

the events in nature through the threshold mechanism. Since data around the threshold may affect the test result, and as a result of many other factors, part of data produced by sensor in the sensing process may be wrong.

Because the definition of the event in the realistic environment is uncertain, and fuzzy-logic based Composite events detection is raised in the literature [21]. Fuzzy logic has had some applications in the wireless sensor network (WSN); literatures [22–24] use fuzzy logic method to select cluster heads, literatures [25, 26] use it to solve the safety problem in wireless sensor network, literature [27] utilizes it in data fusion, and literature [28] imposes it to design routing algorithms. Decision mechanism based on fuzzy logic divides simple events into different levels firstly, calculates the membership of the detected attribute value to each level with membership function, then calculates each rule in the rule base, and finally obtains result with anti-fuzzy.

Objectively, Composite events decision mechanism which is based on fuzzy-logic resolves many unreasonable problems resulting from using threshold value to determine events, but as the decision mechanism needs huge rule base, and rule base will grow exponentially with the number of attributes to be detected increase, thus, it puts forward higher requirements for the sensor node's storage capacity.

In this paper, the proposed fuzzy-decision based Composite event detection algorithm (FL-CED) consists of three parts: first of all, building network topology based on connected dominating sets; the dominant point is local decision node; then using space-time correlation to cluster the perceived value of the same attribute, screening out the correct data, and identifying the fault node; finally, determining Composite events with decision-making mechanism based on fuzzy-decision. The construction of the network topology based on connected dominating set forms the spanning tree which regards the control point as the decision point, and clustering algorithm ensures the correctness of the data involved in determining. Because decision-making mechanism based on fuzzy-decision does not need the support of huge rule base, it is more practical than the decision-making mechanism based on fuzzy logic.

## 2. Network Model

Let  $N$  sensor nodes randomly deploy in the detection area  $A$ , so that the monitoring area is substantially covered and each node is integrated by one or more sensors. Different sensors have different sensing ranges and thus have a different perception and the communication radius is adjustable. Maximum communication radius is described as  $R_C$ . The network consisting of nodes is connected; each node with a unique ID determines its own position by localization algorithm [29, 30] or GPS [31].

Generally, the signal energy transmitted from sensor node is decreasing as the distance from the signal source is increasing. Assuming that  $d_i$  is the distance between node  $i$  and the event source, and  $S$  is the signal energy of the detection node, the lost energy is  $s_i = S \times w(d_i)$ , and  $w(0) = 1$ ,  $w(\infty) = 0$ , and  $w(x) = \theta(x^{-k})$ , where  $w(x)$  is the

signal attenuation function, and it is subject to climatic and other environmental conditions;  $k$  is loss index of the signal transmission, and its range is from 2.0 to 5.0. In this paper, the signal attenuation function [32] is as follows:

$$w(x) = \frac{1}{1 + x^k}. \quad (1)$$

Errors of sensing data of the sensor mainly come from the hardware or the random ambient noise.  $H_0$  and  $H_1$  denote “event does not occur” and “event occurs,” respectively.  $x_i$  represents the measured data of node  $i$ :  $H_0: x_i = n_i$ ;  $H_1: x_i = s_i + n_i$ .  $n_i$  is the noise energy received by the node  $i$ . Let noise  $n_i$  be distributed normally at each node  $i$ ,  $n_i \sim N(\mu, \sigma^2)$ , where  $\mu$  and  $\sigma$  are the mean and variance, respectively. The noise  $\{(n_i \mid \forall i)\}$  is independent of the sensor and obeys normal distribution. The sensing data of the detection node is a normal distribution:  $x_i \mid H_1 \sim N(s_i + \mu, \sigma^2)$ . Due to the independence of the noise, the energy of the received signal of different nodes is different, but the detected value  $\{x_i \mid \forall i, H_1\}$  of the sensor nodes is independent of the space. The peak value of the signal to noise ratio (SNR) is defined as  $\delta = s/\sigma$  to quantify the level of the noise. The signal attenuation and noise model that obeys independent normal distribution has been widely applied in multisensor node signal detection and has been shown [32]. In fact, noise transmission is an approximate normal distribution. The model (i.e.,  $s, w(x), \mu$ , and  $\sigma^2$ ) can be used to estimate the sensing data collected by the WSNs.

### 3. Definitions

*Definition 1* (perceived neighbors).  $V$  is the set of all nodes,  $u, v \in V$ ; let  $v$  be a sensor  $C_i$  with a sensing property  $A_i$ . The perception radius of  $C_i$  is  $R_{S_i}$ . If  $d(u, v) \leq R_{S_i}$ ,  $(d(u, v)$  represents the Euler distance between  $u$  and  $v$ ), then  $v$  is the  $A_i$ -aware neighbor of  $u$ .  $v$  is the  $A_i$ -aware neighbor of  $u$ , which means  $v$  can sense attribute  $A_i$  of  $u$ .

*Definition 2* (active sensing neighbors). The nodes that are not greater than  $R_{S_i}/\eta$  ( $\eta$  is an integer greater than 1) distance from node  $u$  are called the effective  $R_{S_i}$  perception neighbor of  $u$ , and the set of all the points satisfying the properties is denoted by  $N_i(u)$  (the selection of  $\eta$  will be discussed in Section 8.3).

*Definition 3* (simple event). The event occurred can be determined by detecting an attribute of the environment, while the event is called a simple event, denoted as

$$e = \begin{cases} H_0, & P(x) = \text{false} \\ H_1, & P(x) = \text{true}, \end{cases} \quad (2)$$

where  $e$  means a simple event and  $x$  means the corresponding environment attribute value of event  $e$ . When its value is true, it indicates that the event occurred. When the value is false, it indicates that the event did not occur.

*Definition 4* (composite event). Only detecting multiple attributes of the environment can determine whether an

event occurs; then the event is called Composite event, denoted as

$$E = \begin{cases} H_0, & P(x_1, x_2, \dots, x_m) = \text{false} \\ H_1, & P(x_1, x_2, \dots, x_m) = \text{true}, \end{cases} \quad (3)$$

where  $E$  represents the Composite event,  $x_1, x_2, \dots, x_m$  represent  $m$  kinds of attributes values of the environment, and  $P(x_1, x_2, \dots, x_m)$  is the decision function, whose meaning is the same with the simple event.

*Definition 5* (event area). If detecting events (simple event or Composite event) in any position of area  $A$ , the results are all  $H_1$ , then area  $A$  is called the event area.

In the boundary of the event area, since attribute values of the environment do not vary, the environmental attribute values detected by the node can approximately reflect the state of the attribute in the node sensing range. Because of harsh environmental conditions, nodes are likely to be damaged. Then we can deploy many nodes of the same type to detect the region. Based on the  $k$ -watching nature of the event in literature [10], we propose a regional  $k$ -watching concept.

*Definition 6* ( $k$ -watched area). If all attributes  $A_1, A_2, \dots, A_m$  of the environment are detected by the node integrated by  $K$  sensors  $C_1, C_2, \dots, C_m$ , respectively, the region is  $k$ -watched, and the region also meets the  $k$ -watching nature. If  $N_i(u)$  ( $1 \leq i \leq m$ ) of node  $u$  in the monitoring area has  $K$  nodes integrated  $C_i$ , respectively, then  $u$  is said to be  $k$ -watching decision node, referred to as a decision node,  $R_d = (1 - 1/\eta) \min(R_{S_1}, R_{S_2}, \dots, R_{S_m})$  is called the  $k$ -watching policy area, referred to as decision-making scope.

*Definition 7* (detecting voids). If a node is not the decision node, and it is not in a decision-making scope of a decision node, then the events surrounding the node could not be detected; the location of the node is called a detection void.

*Definition 8* (clustering [33]). Let  $X$  be a data set,  $X = \{x_1, x_2, \dots, x_k\}$ ;  $x_i$  denotes properties of object  $i$ . An  $m$ -cluster of  $X$  is denoted as  $\mathfrak{R}$ ; then  $X$  is divided into  $m$  classes  $C_1, C_2, \dots, C_m$  by  $\mathfrak{R}$  and satisfies the following three conditions:

- (1)  $C_i \neq \emptyset, i = 1, \dots, m$ ;
- (2)  $\cup_{i=1}^m C_i = X$ ;
- (3)  $C_i \cap C_j = \emptyset, i \neq j, i, j = 1, \dots, m$ .

*Definition 9* (membership function). An arbitrary mapping from the domain  $X$  to the closed interval  $[0, 1]$ :  $u_{\bar{A}} : X \rightarrow [0, 1]$ , for any  $x \in X, x \rightarrow u_{\bar{A}}(x), u_{\bar{A}}(x) \in [0, 1]$ ; then  $\bar{A}$  is called a fuzzy subset of  $X, u_{\bar{A}}$  is called membership function of  $x$ , and  $\mu_{\bar{A}}(x)$  is called the membership of  $x$  to fuzzy set  $\bar{A}$ , denoted as

$$\bar{A} = \{(x, u_{\bar{A}}(x)) \mid x \in X\}. \quad (4)$$

Given a limited domain  $X = \{x_1, x_2, \dots, x_m\}$ , its fuzzy subset  $\bar{A}$  can be denoted as Chad notation:

$$\bar{A} = \frac{u_{\bar{A}}(x_1)}{x_1} + \frac{u_{\bar{A}}(x_2)}{x_2} + \dots + \frac{u_{\bar{A}}(x_i)}{x_i} + \dots + \frac{u_{\bar{A}}(x_n)}{x_n}, \tag{5}$$

where  $x_i \in X$  ( $i = 1, 2, \dots, n$ ) is the element of the domain,  $\mu_{\bar{A}}(x_i)$  is the membership function of  $X$  to  $\bar{A}$ ,  $0 \leq \mu_{\bar{A}}(x_i) \leq 1$ , the formula represents a fuzzy set of  $n$  elements, and “+” is sign for Chad mark.

### 4. Criteria

Assuming that detecting Composite event  $E$  needs to detect environmental attributes  $A_1, A_2, \dots, A_m$ , these properties are detected by the integrated sensor  $C_1, C_2, \dots, C_m$ , corresponding perception range is  $R_{s1}, R_{s2}, \dots, R_{sm}$ , respectively, and  $x_1, x_2, \dots, x_m$  are the values of corresponding attribute. The node that detects  $P(x_i) = true$  sends the message (attribute, value) to the local decision node, and the concrete properties values are not contained in the message, which only contains 0 or 1; 1 shows that  $P(x_i) = true$  and can reduce the length of data packets. Local decision node operates after receiving packets and judges whether event happens within the decision scope.

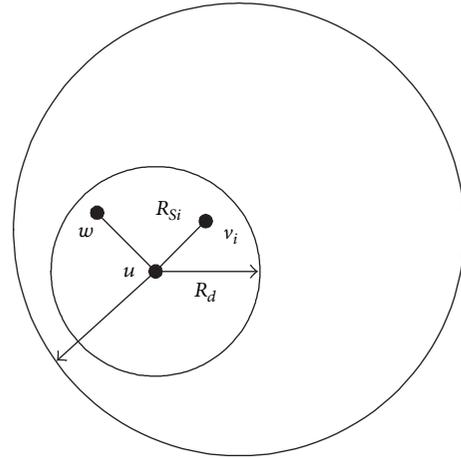
*Criteria.* For any node  $u$  in the network,  $N_1(u), N_2(u), \dots, N_m(u)$  have the node with integrated sensors  $C_1, C_2, \dots, C_m$ , and their readout are  $x_1, x_2, \dots, x_m$ ; if  $P(x_1, x_2, \dots, x_m) = true$ , Composite event could happen in the circular area, which regards  $u$  as the center and  $R_d = (1 - 1/\eta) \min(R_{s1}, R_{s2}, \dots, R_{sm})$  as radius; the set of all nodes within decisions scope of  $u$  is denoted as  $N_D(u)$ .

**Theorem 1.** For any node  $u$  in the network, one can choose an node integrated sensor  $C_1, C_2, \dots, C_m$  from  $N_1(u), N_2(u), \dots, N_m(u)$ , a collection constituted of these nodes is recorded as  $\{v_1, v_2, \dots, v_m\}$  (elements can be repeated), and if there is such a  $\{v_1, v_2, \dots, v_m\}, \{v_1, v_2, \dots, v_m\}$  can sense any environment attributes of any position in the decision scope of  $u$ .

*Proof.* Take a  $w$  within the decisions scope of  $u$  and decision radius of  $u$  is  $R_d$ . Assuming that the distance  $d(u, v)$  between  $u$  and  $v$  is less than  $R_d$  and the distance  $d(u, v)$  between  $u$  and node  $v_i$  integrated  $C_i$  is less than  $R_{Si}/\eta$ , see Figure 1 then

$$\begin{aligned} d(w, v_i) &\leq d(w, u) + d(u, v_i) \\ &\leq R_d + \frac{R_{Si}}{\eta} \\ &\leq \left(1 - \frac{1}{\eta}\right) R_{Si} + \frac{R_{Si}}{\eta} = R_{Si}. \end{aligned} \tag{6}$$

Thus, environment attributes  $A_i$  of  $w$  can be tested by  $v_i$ . Since  $w$  and  $i$  are arbitrary,  $\{v_1, v_2, \dots, v_m\}$  can sense any environment attribute within any position of decision scope of  $u$ .



$R_{Si}$ : Sensing radius of  $C_i$   
 $R_d$ : Decision radius

FIGURE 1: Decision making range diagram.

Theorem 1 shows that as long as selected part of decision nodes is in the monitoring area and other nodes are within the scope of their decisions, then we can determine the status of the incident in this monitoring area by testing the environment with these decision nodes. However, attribute values received by local decision node not necessarily reflect changes of the environment correctly. Decision mechanism proposed in this paper is divided into two parts: first, the local decision nodes process the detecting data with clustering algorithm and use the processed data to make fuzzy decision and get the final conclusion. □

### 5. Construction of the Network Topology

Abstracting the network into a graph  $G = (V, E)$ ,  $G' = (V', E')$  and  $G_D = (V, E_D)$ ,  $V$  represents the set of all the nodes in the network,  $E$  represents the set consisted of neighbor node pairs communicating with each other in  $V$ , and  $V' = V + \{\text{sink}\}$ , where sink is the base station.  $E'$  represents the set consisted of neighbor node pairs communicating with each other in  $V'$ ;  $E_D$  is the set consisting of node pairs whose Euler distance is not more than  $R_d$ . The CETC algorithm proposed in this paper regards dominating sets as the topology of the event detection and selects a subset of the graph  $G$  as  $V_D$ , where the point is the local decision node. The choice of  $V_D$  should meet the following demands.

- (1) Nodes in  $V_D$  are decision nodes.
- (2)  $V_D$  is a dominating set of  $G_D = (V, E_D)$ ; domination relationship is the decision making.
- (3) Use a distributed algorithm, each node can decide whether to be a point of  $V_D$  in local scope.
- (4) Algorithm has lower time complexity and message exchange complexity.

- (5) Nodes in  $V_D$  have more residual energy than other nodes.
- (6) The number of points in  $V_D$  should be less as possible and these points are scattered in space.

Based on the goal above, the process of building the topology structure of the network is divided into two steps: selecting  $V_D$  and  $V_s$ . Selecting  $V_D$  is selecting a dominating set of  $G_D$ . Selecting  $V_s$  is selecting part of the nodes in  $V$ , providing the nodes in  $V_D$  with attributes that they cannot sense, and making the decision range of the node in the  $V_D$  satisfy the  $k$ -watching nature. Initially, there are four types of nodes in the network.

- (1) Sleep node (SLEEP): node inactive.
- (2) Local decision node (DECIDE): nodes collecting the information of peripheral nodes and determining whether the Composite events happen within  $R_d$ .
- (3) Sensor nodes (SENSE): perceiving attribute values of the environment and sending them to the local decision node.
- (4) Connect nodes (CONNECT): nodes forwarding results of the event.

Local decision node has three decision-making results (RESULT): decision scope is in the area of the event, decision scope is on the boundary of the event area, and no events are in the decision scope.

Initially, all nodes are SLEEP, and the network performs CETC algorithm, execution time is  $T_c$ , when the algorithm ends, nodes start to detect the event in the monitoring area, and the time of each round is the  $T_R$ .

(1) *Selecting  $V_D$ .* First of all, determine which nodes can be  $k$ -watching decision nodes in the network. Assume that node  $u$  integrated a variety of sensors  $C_1, C_2, \dots, C_l$ , so  $u$  with  $R_{si}/\eta$  ( $1 \leq i \leq l$ ) for the radius broadcast decision node to discover data packets (LDN packet), which includes the ID of the node, the type of sensor  $C_i$  integrated by the node, and the residual energy of the node. After time  $T_D$ , each node receives the LDN data packet from the neighboring nodes, extracts the information of LDN packets, and stores it. Each node will determine whether received at least  $k$  LDN packets sent by nodes integrated of  $C_i$  ( $1 \leq i \leq m$ ); if so, then the node will be the  $k$ -watching decision node; the set which consists of  $k$ -watching decision nodes is denoted as  $V_{CBD}$ .

Then, selecting a part of nodes from  $V_{CBD}$  to constitute  $V_D$  so that  $V_D$  is a dominating set of the graph  $G_D$ , nodes of  $V_D$  will be the local decision nodes. Local decision node has more residual energy compared to other nodes, and there should be more nodes within its decision scope, namely, greater degree. The degree of the node means the number of neighbor nodes; the degree of the node  $u$  in figure  $G$  is  $|N_G(u)|$ . Choosing the node with greater degree contributes to reducing the number of local decision nodes in network and making the distribution of local decision nodes more dispersed. The priority  $Prio$  represents the residual energy of

the node and the whole situation of the degree; the calculation of the priority of  $u$  is given by

$$Prio_u = \frac{E_{residual}}{E_{max}} \times \frac{d_u}{\bar{d}}, \tag{7}$$

where  $E_{residual}$  denotes the remaining energy of the node,  $E_{max}$  is the initial energy of the node,  $d_u$  is the degree of the node, and  $\bar{d}$  indicates the average degree of the nodes in graph  $G_D$ . Consider

$$\bar{d} = \left\lfloor (N - 1) \times \frac{\pi R_d^2}{A} \right\rfloor, \tag{8}$$

where  $N$  is the number of nodes in a network and  $A$  is the size of the monitoring area. Node broadcasts neighbor discovery packets (LN packets) with radius  $R_d$  after confirming whether the node itself is  $k$ -watching decision node; the message contains the ID of the node and whether it is  $k$ -watching node. After time  $T_N$ , all  $k$ -watching decision nodes received LN packets sent from nodes within the scope of  $R_d$ . In this way, each of the  $k$ -watching decision nodes will know its own degrees; then the node counts their priority  $Prio$ .

$k$ -watching decision nodes broadcast the priority packet (PP packet) with  $R_d$  radius; the priority information is contained in PP data packet. After  $T_P$  time, each node receives packets sent by the  $k$ -watching decision nodes in the range of  $R_d$ .  $k$ -watching decision nodes extract the information of all nodes in PP data packets and then calculate the waiting time of sending BDN packets. The waiting time of a  $k$ -watching decision node  $u$  before sending BDN packets is  $t_u$ :

$$t_u = \begin{cases} 1 - \frac{\lfloor ((Prio_u - Prio_{min}) / (Prio_{max} - Prio_{min})) \times L \rfloor}{L} \\ \quad \times T_\tau + T_{random}, & U \text{ received PP packets} \\ T_{random}, & U \text{ did not receive PP packets,} \end{cases} \tag{9}$$

where  $Prio_{max}$  is the maximum priority among the PP packet received by  $u$  and  $Prio_{min}$  is the minimum priority among the PP packet received by  $u$ .  $T_\tau$  is a given time period,  $T_\tau$  is divided into  $L$  segments, and  $T_{random}$  is a random moment in  $[0, T_\tau/L]$ . If  $u$  received a PP packet, the wait time is calculated by the first equation; if  $u$  does not receive a PP packet,  $u$  will choose a random value in  $[0, T_\tau/L]$  as the waiting time. After time  $T_s$ , all nodes have finished the calculation and start timing. Waiting for time  $t_u$ ,  $u$  will send BDN packets. During the waiting period, if  $u$  received the BDN data packet from other nodes, it will give up sending operation. Otherwise, the node sends a BDN packet and informs that it will become the local decision node and set its STATE to DECIDE.

If a non- $k$ -watching decision node  $v$  does not accept BDN data packets sent from other nodes in time  $(1 + 1/L)T_\tau$ , but received PP data packet before, it will send passive local

decision node data packets (CDN data packet) to node  $w$  of maximum priority in PP data packets and inform  $w$  to be the local decision node;  $w$  will set the STATE to DECIDE after receiving the message. If being not received PP packet, then  $v$  is not in the scope of decision node of  $k$ -watching decisions nodes; thereby there is a detect void in position  $v$ .

(2) *Selecting  $V_S$ .* As local decision nodes have integrated limited sensors and need to select the surrounding nodes as the sensing nodes to provide the environmental information that could not be detected, there is a sensor node list in the cache of the local decision node, storing related information of the sensor node and containing ID of the node and property  $A_i$ . When  $P(x_i) = true$ , the sensing nodes will send simple events data packet (SE packet) to the local decision node in the list.

When choosing  $V_D$ , each local decision node has received the LDN data packet sent from the neighboring nodes. Local decision node  $v$  extracts node information from the received LDN data packet and merges the same information. If node  $u$  can provide information  $A_1$  and information  $A_3$ , then combine the two kinds of information to the perception information of node  $u$ . Each kind of sensor  $C_i$  ( $1 \leq i \leq m$ ) has a counter count <sub>$i$</sub> ; initially, count <sub>$i$</sub>  =  $k$ . The value of the corresponding count <sub>$i$</sub>  ( $j \leq i \leq l$ ) of the integrated sensor will be minus 1 by  $v$ ; then  $v$  will sort node information in the LDN data packet by residual energy of the node from large to small. Selecting nodes from the sorted sequence sequentially, let the node selected currently be  $u$ ; if the corresponding count <sub>$i$</sub>  of the integrated sensor of  $u$  is larger than 0, then  $v$  will choose  $u$  as the sensing node to provide the information of attribute  $A_j$ , and the corresponding count <sub>$i$</sub>  is minus 1. If the corresponding count <sub>$i$</sub>  of all sensors integrated by  $u$  is 0, then  $v$  will not select  $u$  as the sensor node. All of the count <sub>$i$</sub>  have become 0,  $v$  will cease to choose. When node  $v$  has selected the sensor node,  $v$  will send data packets (SN packet) to the selected sensing node; SN data package includes node ID and the list  $A_1, A_2, \dots, A_l$  of attributes to be detected. The sensor node extracts the information from the received SN data packet, adds the source node sending a SN packet to its local decision node list, and then returns the acknowledgment packet (CF packet). Local decision node will add the information of the sensor node to its local decision node list after receiving a confirmation message. Let  $T_{SS}$  be the time required for selecting sensor nodes. The pseudocode of CETC algorithm is as shown in Algorithm 1.

In the initial conditions, the RESULT of the local decision node is Out\_Event. If there is part of  $e_i = 1$ , but  $E = 0$ , it indicates the decision range of the local decision node on the boundaries of the Composite event and sets the RESULT of the local decision node as Boundary\_Event. If  $E = 1$ , set the RESULT of local decision node as In\_Event. If the RESULT is not Out\_Event, the local decision node will send event data packets (CE) to the base station; the data packets contain the position and the RESULT of the local decision node. The base station will determine the area of Composite event according to the positions and decision contained in the received CE data packet.

## 6. Cluster Analysis and Fault Detection

Supposing the transmit period of perception nodes is  $T_s$ , the local decision node receives the data transmitted by the perception node periodically. It needs to detect  $m$  kinds of attributes of the environment in order to determine the Composite events. The decision ranges of local decision making nodes satisfy  $k$ -watching nature; that is to say, the local decision node will receive  $m \times k$  data each cycle. We use the sliding window mechanism, making the local decision node stores the data of the recent  $p$  cycles at any moment. That means the size of the sliding window is  $p$ , so it saves  $p \times m \times k$  data each moment. The value of  $j$ th attribute at time  $i$  ( $1 \leq i \leq p$ ) of the sliding window is  $A_j^i = \{x_{j1}, x_{j2}, \dots, x_{jk}\}$ , the data received at time  $i$  ( $1 \leq i \leq p$ ) is  $A^i = \{A_1^i, A_2^i, \dots, A_m^i\}$ , and the data of the  $j$ th attribute of the local decision node is  $A_j = \{A_j^1, A_j^2, \dots, A_j^p\}$ . Data received is  $A = \{A^1, A^2, \dots, A^p\}$  and carrying out cluster analysis on  $A_j^i$  and  $A_j$ , algorithm including the two clustering analysis is called two-dimensional  $\tau$ -GAS algorithm. GAS (generalized agglomerative scheme) [34] is a kind of hierarchical clustering mechanism; two-dimensional  $\tau$ -GAS algorithm adds restriction conditions on the basic of the GAS algorithm and uses spatiotemporal correlation to cluster the data.

Initially, each attribute value corresponds to a class, a total of  $k$  classes, as  $\mathfrak{R}_0 = \{C_l = \{x_l\}, l = 1, 2, \dots, k\}$ ; the classification obtained at step  $q$  is denoted as  $\mathfrak{R}_q$ . Then, the clustering of the first layer data is carried out. Due to spatiotemporal correlation, we mark the differences between the two classes using the poor between the two elements with the smallest difference in two classes, denoted as  $g(C_n, C_q)$ :

$$g(C_n, C_q) = \min(|x_i - x_j|), \quad (x_i \in C_n, x_j \in C_q). \quad (10)$$

Then merge the smallest two classes of  $g(C_n, C_q)$  and get  $C_r = C_n \cup C_q$ . Using span  $\psi$  to denote the maximum difference between the elements in a class as follows:

$$\psi_{C_r} = \max(|x_i - x_j|), \quad (x_i, x_j \in C_r). \quad (11)$$

Values of the same attribute are similar; therefore, for the  $j$  attribute, given a threshold  $\tau_j$ , and  $\psi_{C_r} \leq \tau_j$ ,  $\tau_j$  could not be too small, in case that reasonable data are neglected,  $\tau_j$  could not be too large, in case that wrong data and correct data are in a class. When  $\psi_{C_r} \leq \tau_j$ , merge two minimum classes of  $g(C_n, C_q)$  and get  $\mathfrak{R}_1 = (\mathfrak{R}_0 - \{C_n, C_q\}) \cup \{C_r\}$ . Then repeat the above steps until merging of any two of the classes does not satisfy the formula  $\psi_{C_r} \leq \tau_j$ , or leaving only one classify finally.

Setting the class  $C_e$  with maximum number of elements as a correct data class, the average  $\bar{x}_j^i$  is the value of attribute  $j$  at time  $i$ :

$$\bar{x}_j^i = \frac{1}{|C_e|} \sum_{x_i \in C_e} x_i. \quad (12)$$

If the number of category  $C_e$  with maximum number of elements is more than one, then calculate  $\bar{x}_j^i$  of each

Input: graph  $G = (V, E)$ , graph  $G_D = (V, E_D)$ ,  $\forall u \in V$  output:  $V_D, V_S$   
 Initialization: STATE = SLEEP, CB\_Decide = false, CB\_Decide indicates whether the node is a  $k$ -watching decision node  
 Choose  $V_D$ :

- (1) Broadcast LDN packets //1 is performed in the period of time  $T_D$
- (2) Receive LDN packet, calculate itself whether can be a  $k$ -watching decision node according to the condition of the determine criterion. If so, CB\_Decide = true. //(2)–(4) will be implemented in the  $T_N$  time period
- (3) broadcast(LN packets);
- (4) receive(LN packets);
- (5) if (CB\_Decide = true) //(5)–(10) performed in the  $T_P$  time period
- (6)     calculate priority  $Prio_u$ ;
- (7)     broadcast(PP packets);
- (8)    end if
- (9) receive(PP packets); //Before the end of the  $T_P$  time period, the node is receiving PP data packet
- (10) if (CB\_Decide = true)
- (11)    Calculate  $t_u$ ; //11 performed in the  $T_s$  time period
- (12)     $t$  starts timing after the period of time  $T_S$ ; //(14)–(22) performing in  $(1 + 1/L)T_t$  time period
- (13)    while ( $t < t_u$ )
- (14)     if (receive(BDN packets))
- (15)       stop the plan of sending BDN data package;
- (16)     end if
- (17)    end while
- (18) if (!receive(BDN packet)) //The node not received the BDN packet
- (19)    broadcast(BDN packet);
- (20)    STATE = DECIDE;
- (21)    end if
- (22) end if
- (23) if (!receive(BDN packet) && CB\_Decide = false) //(24)–(34) Performed in the  $T_Q$  time period
- (24)     $v \leftarrow$  the maximum  $Prio$  node in PP packets;
- (25)    if ( $v$  exists)
- (26)     send( $v$ , CDN packet);
- (27)    else
- (28)     detected voids exist;
- (29)    end if
- (30) end if
- (31) if (receive(CDN packets))
- (32)    STATE = DECIDE;
- (33) end if

Choose  $V_S$ : //The process executes in time period  $T_{SS}$

- (1) if (STATE = DECIDE)
- (2)    extract the information of LDN data packet;
- (3)    Count <sub>$i$</sub>  =  $k$ ;
- (4)    corresponding count <sub>$i$</sub>  of all sensors of the current node = count <sub>$i$</sub>  - 1;
- (5)    sort node information by residual energy from large to small and get set  $U$ .
- (6)    foreach ( $u$  in  $U$ )
- (7)     if (any count <sub>$i$</sub>  = 0)
- (8)      break;
- (9)     end if
- (10)    if (count <sub>$i$</sub>  corresponding to the integrated sensor on  $u > 0$ )
- (11)     select  $u$  as the sensor node, let the test attribute  $A_i$ , count <sub>$i$</sub>  = count <sub>$i$</sub>  - 1;
- (12)    else
- (13)     continue;
- (14)    end if
- (15) end foreach
- (16) get a sensing node set  $\{u_1, u_2, \dots, u_p\}$ , And property  $u_i$  needs to  $\{A_1, A_2, \dots, A_p\}$ ;
- (17) foreach ( $u_i$  in  $\{u_1, u_2, \dots, u_p\}$ )
- (18)    send( $u_i$ , SN data packet);
- (19) end for each
- (20) end if
- (21) if (receive(SN packets))
- (22)    add the SN information to the local decision node list;
- (23)    send(SN.source, CF data packet);

```

(24)  if (STATE ≠ DECIDE)
(25)    STATE = SENSE;
(26)  end if
(27) end if
(28) if (receive(CF data packet))
(29)  extract the contents of the CF data package, add it to the sensing node list;
(30) end if
 $V_D = \{v|v. STATE = DECIDE\};$ 
 $V_S = \{v|v. STATE = SENSE\};$ 

```

ALGORITHM 1: Construction algorithm of topology (CETC).

TABLE 1: Temperature data.

$p$	$k$					
	1	2	3	4	5	6
1	53.8	55.1	55.8	54.3	78.1	55.9
2	55.1	54.9	54.1	55.5	54.1	56.4
3	55.7	81.2	54.1	55.6	21.6	55.1
4	28.9	30.7	10.1	29.7	27.8	30.5
5	54.3	53.2	55.3	54.8	55.8	53.8

classification; finally, take values which are most close to event attribute, and the corresponding class will be attributed to the correct category. As classifying temperature data in fire detection, if the number of category  $C_e$  with maximum number of elements is more than one, then the class  $C_e$  with the largest  $\bar{x}_j^i$  is considered as a correct category. The average of all the  $j$ th properties at  $p$  moments is  $\bar{A}_j = \{\bar{x}_j^1, \bar{x}_j^2, \dots, \bar{x}_j^p\}$ . In order to eliminate the temporary failure of the node, cluster the data of  $\bar{A}_j$  by the process above and get the average  $\bar{x}_j$ . The algorithm above is called two-dimensional  $\tau$ -GAS algorithm; the description of the algorithm is as shown in Algorithm 2.

Next, we present an example of performing a two-dimensional  $\tau$ -GAS cluster on temperature. Table 1 shows a set of temperature data when  $p = 5$ ,  $k = 6$ .

Analyzing the group of data with two-dimensional  $\tau$ -GAS algorithm. According to the algorithm, it will analyze the data of every moment. Next we will process the data at the third moment:

$$A_t^3 = \{55.7, 81.2, 54.1, 55.6, 21.6, 55.1\}, \quad \tau_t = 10^\circ\text{C}.$$

$$\begin{aligned}
& A_t^3 \{ \{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\} \} \\
& \mathfrak{R}_0 \{ \{x_5\}, \{x_3\}, \{x_6\}, \{x_4\}, \{x_1\}, \{x_2\} \} \\
& \mathfrak{R}_1 \{ \{x_5\}, \{x_3\}, \{x_6\}, \{x_4, x_1\}, \{x_2\} \} \\
& \mathfrak{R}_2 \{ \{x_5\}, \{x_3\}, \{x_6, x_4, x_1\}, \{x_2\} \} \\
& \mathfrak{R}_3 \{ \{x_5\}, \{x_3, x_6, x_4, x_1\}, \{x_2\} \}.
\end{aligned} \tag{13}$$

Figure 2 is the clustering process of  $A_t^3$  with three steps in total; we get  $\mathfrak{R}_0$ ,  $\mathfrak{R}_1$ ,  $\mathfrak{R}_2$ , and  $\mathfrak{R}_3$  after clustering. The number of elements in  $\{x_2, x_6, x_4, x_1\}$  of  $\mathfrak{R}_3$  is the

largest; therefore,  $x_2$  and  $x_5$  are considered to be unreliable;  $\bar{x}_t^3 = 55.13$ . Analyzing the other data, we get  $\bar{A}_t = \{54.98, 55.02, 54.98, 55.02, 54.53\}$ . Then clustering  $\bar{A}_t$  again, we will get the class  $\{54.53, 54.98, 55.02, 55.13\}$  with the most elements, 29.52 may be caused by the temporary changes of the environment or temporary failure of the node. Therefore,  $\bar{x}_t = 54.92$  is used to indicate the current temperature of the environment.

The  $\tau$ -GAS function will sort the collected data first; the complexity of the algorithm is  $O(k \log k)$ . In the layer  $t$ , it is needed to compare for  $k-t-1$  times; the comparison number of the algorithm is

$$\sum_{t=0}^{k-1} (k-t-1) = \frac{k(k-1)}{2}. \tag{14}$$

Then, the total time complexity of the two-dimensional  $\tau$ -GAS algorithm is  $O(p \times k^2 + p^2)$ .

As for  $A_j^i = \{x_{j1}, x_{j2}, \dots, x_{jk}\}$ , if  $C_e = \{x_{jq}, x_{j2}, \dots, x_{j\ell}\}$ , then the node producing the data  $\{x_{j1}, x_{j2}, \dots, x_{j(q-1)}, \dots, x_{j(\ell+1)}, \dots, x_{jk}\}$  is considered to fail. The node will be informed by the local decision node. Before running CETC algorithm, node sets  $F_i = 0$  for each sensor,  $F_i$  represents the times of failure of the node in a round. If local decision node determines that a sensory data is invalid, then it will notify the corresponding nodes, value  $F_i$  of the corresponding node increases 1. Time of each round is  $T_R$ , the execution time of CETC algorithm is  $T_C$ , and data collection cycle of local decision node is denoted by  $T_E$ ; then the collecting times of each round are

$$N = \frac{T_R - T_C}{T_E}. \tag{15}$$

The perception probability of a sensor node failure is

$$p_f = \frac{F_i}{N} = \frac{F_i \times T_E}{T_R - T_C}. \tag{16}$$

Given a threshold  $\theta$ , if  $p_f > \theta$ , the node cannot sense the environment.

## 7. Fuzzy Decision of Composite Events

Fuzzy decision is an important part of fuzzy system theory. A decision is the countermeasures and strategies taken on

```

Input:  $A_j$ 
output:  $\bar{x}_j$ 
For ( $i = 1$  to  $p$ )
     $\bar{x}_j^i = \tau\text{-GAS}(A_j^i)$ ;
End for
 $\bar{A}_j = \{\bar{x}_j^1, \bar{x}_j^2, \dots, \bar{x}_j^p\}$ ;
 $\bar{x}_j = \tau\text{-GAS}(\bar{A}_j)$ ;
Function  $\tau\text{-GAS}(\{x_1, x_2, \dots, x_k\})$ 
Initialize:
    Sort the data of  $\{x_1, x_2, \dots, x_k\}$  from small to large
    Choose  $\mathfrak{R}_0 = \{C_i = \{x_l\}, l = 1, 2, \dots, k\}$  as the initial classification
     $t = 0$ 
Repeat:
     $t = t + 1$ ;
     $g(C_i, C_j) = \min g(C_n, C_q) (\psi_{C_r} \leq \tau_j, C_r = C_n \cup C_q)$ ;
     $\mathfrak{R}_t = \mathfrak{R}_{t-1} - \{C_i, C_j\} + C_r$ ;
Until (merger of any two class satisfies  $\psi_{C_r} > \tau_j$  or  $|\mathfrak{R}_t| = 1$ )
     $C_e = C_j, |C_i| = \max(|C_\ell|), C_\ell \in \mathfrak{R}_t$ ;
     $\bar{x} = \frac{1}{|C_e|} \sum_{x_i \in C_e} x_i$ ;
Return  $\bar{x}$ ;
End function  $\tau\text{-GAS}$ 
    
```

ALGORITHM 2: Two-dimensional  $\tau$ -GAS algorithm.

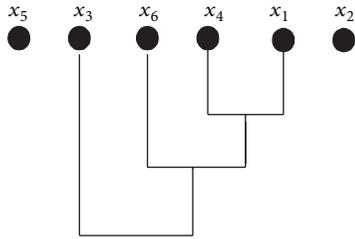


FIGURE 2: Clustering process.

a thing [35]. In real life, factors that affect the determine results of Composite events are fuzzy mostly, a given method is sorted by fuzzy decision algorithm; decision results can be considered as solution set, from which the most appropriate processing scheme is selected. Such as in fire detection, the decision results of the node are divided into three possible schemes send information of event occurs, send information of the abnormal environment, without any operation. In some cases, part of the property value of the environment is higher, but the event has not occurred; then the scheme of abnormal environment was sent. For this kind of abnormal phenomenon, we will process it additionally.

The estimate value  $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m\}$  of the current attribute of the environment can be obtained by the two-dimensional  $\tau$ -GAS algorithm; with the estimated value, we can obtain the optimal scheme of the event decision. In the paper, we adopt a compromised type of fuzzy multiattribute method [36] to make decisions on events; its basic idea is that for the initial sample data, virtual fuzzy positive ideal solution, and fuzzy negative ideal solution, the fuzzy positive ideal solution

is constituted of the maximum value of the fuzzy parameter values of each attribute, and the fuzzy negative ideal solution is constituted of the minimal value of fuzzy indicator values. Then calculate the weighted distance between each alternative and the fuzzy positive ideal solution and fuzzy negative ideal solution with weighting method. On this basis, calculate the membership degree of each alternative belonging to fuzzy positive ideal; the greater the degree of membership, the more ideal the scheme.

The paper modifies the compromised type of fuzzy multiattribute method appropriately to adapt the determination of the Composite event in the wireless sensor network. The set of the attributes of the determining solution is  $V = \{V_1, V_2, \dots, V_m\}$ , the set of node decision scheme is  $A = \{A_1, A_2, \dots, A_n\}$ , elements of  $A$  are ordered,  $V_i$  is the factor that the scheme needs to consider, such as fire detection, and the  $V_i$  can be temperature; it can also be a temperature of fixed period of time.  $A_i$  indicates node processing solution: choosing the optimal solution by decision algorithm. The following is the main steps of the determining mechanism based on the fuzzy decision.

Step 1. The membership degree function matrix of each attribute relative to the scheme is  $I = (u_{ij}(x))_{n \times m}$ , and calculate the fuzzy indicators matrix  $F = (f_{ij})_{n \times m}$ :

$$I = \begin{pmatrix} u_{11}(x) & u_{12}(x) & \cdots & u_{1m}(x) \\ u_{21}(x) & \cdots & \cdots & u_{2m}(x) \\ \vdots & \vdots & \vdots & \vdots \\ u_{n1}(x) & u_{n2}(x) & \cdots & u_{nm}(x) \end{pmatrix}. \tag{17}$$

$u_{ij}(x)$  is a membership function, indicating the suitability of adopting scenario  $i$  when the value of  $j$ th attribute is  $x$ . Membership function can be formulated by experts of the relevant application domain or obtained by machine learning algorithms. Given  $\{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_m\}$ , calculate the fuzzy indicators matrix  $F$ :

$$F = \begin{pmatrix} u_{11}(\bar{x}_1) & u_{12}(\bar{x}_2) & \cdots & u_{1m}(\bar{x}_m) \\ u_{21}(\bar{x}_1) & \cdots & \cdots & u_{2m}(\bar{x}_m) \\ \vdots & \vdots & \vdots & \vdots \\ u_{n1}(\bar{x}_1) & u_{n2}(\bar{x}_2) & \cdots & u_{nm}(\bar{x}_m) \end{pmatrix}. \quad (18)$$

Step 2. Weigh vector  $w = \{w_1, w_2, \dots, w_m\}$ ; the weight vector  $w$  reflects the influence of different attributes on scheme selection.

Step 3. Construct fuzzy decision matrix  $D = (r_{ij})_{n \times m}$ , where  $r_{ij} = w_j \times f_{ij}$ .

Step 4. Define the fuzzy positive ideal solution  $M^+$  and fuzzy negative ideal solution  $M^-$ , where  $M^+ = (M_1^+, M_2^+, \dots, M_m^+)$  and  $M^- = (M_1^-, M_2^-, \dots, M_m^-)$ . Among this,  $M_j^+ = \max\{r_{1j}, r_{2j}, \dots, r_{nj}\}$ ,  $j = 1, 2, \dots, m$  is the maximum value relative to indicator values of the  $j$ th column. The fuzzy index  $M_j^- = \min\{r_{1j}, r_{2j}, \dots, r_{nj}\}$  ( $j = 1, 2, \dots, m$ ) is the minimal value relative to fuzzy indicator values of the  $j$ th column.

Step 5. Calculate the distance  $d_i^+$  between scheme  $A_i$  and fuzzy positive ideal solution  $M^+$  and the distance  $d_i^-$  between scheme  $A_i$  and fuzzy negative ideal solution  $M^-$ , respectively:

$$d_i^+ = \sqrt{\sum_{j=1}^m (r_{ij} - M_j^+)^2}, \quad i = 1, 2, \dots, n \quad (19)$$

$$d_i^- = \sqrt{\sum_{j=1}^m (r_{ij} - M_j^-)^2}, \quad i = 1, 2, \dots, n.$$

Step 6. Select decision scheme.

Let a scheme  $A_i$  belong to the fuzzy positive ideal with membership  $u_i$ ; here

$$u_i = \frac{d_i^-}{d_i^+ + d_i^-}. \quad (20)$$

Clearly,  $0 \leq u_i \leq 1$ , if the  $A_i$  is more close to  $M^+$ , the  $u_i$  is more close to 1. According to the membership degree, sort  $u_i$  in descending order. The greater the  $u_i$  is, the more reasonable the scheme  $A_i$  is. Scheme of  $A$  is sorted from high to low order according to the urgency; when  $u_i = u_j = \max\{u\}$  ( $i \neq j$ ), if  $i < j$ , select  $A_i$ ; otherwise, select  $A_j$ .

There are two important parameters in the algorithms above: the set of membership functions  $I = (u_{ij}(x))_{n \times m}$  and weight vector  $w$ . The algorithm can be suitable for the Composite event detection in different environments by configuring different  $I$  and  $w$ . Such as fire detection, the environment in summer and winter is different. The

temperature in summer is higher; modifying the membership functions associated with temperature and cut down the weight of temperature can reduce the effect of temperature on scheme selection. The temperature in winter is lower; raising the weight of temperature can improve the effect of temperature on scheme selection.

The decision mechanism based on threshold is unable to process the fuzzy properties. The rule base grows exponentially as the attribute increases in the decision mechanism based on fuzzy logics. The proposed decision mechanism of Composite event based on fuzzy logic can not only deal with the vaguely defined attribute but also with the condition that the number of elements in the membership function matrix  $I$  grows exponentially as  $m$  increases when  $n$  is given.

For scheme  $A_i$ , if part of  $r_{ij}$  is bigger, then  $u_i$  will be larger. The determination mechanism based on fuzzy decision cannot indicate the complex Boolean function in the determination mechanism based on threshold and complex rules in the decision mechanism based on fuzzy logic; thus, it cannot judge the more complex Composite event. When most attributes of the environment or part of properties with greater weight meet the requirements of the occurrence of Composite event, the determination mechanism based on fuzzy decision can determine whether the Composite event occurs, which meets the nature of most Composite events in the real world; the rules package in the literature [21] contains such rules. Therefore, the determination mechanism based on fuzzy decision of the Composite event can be applied to the Composite event detection in the general situation.

## 8. Performance Analysis

This section will discuss the time complexity and the complexity of message exchange of the algorithm and analyze advantages of detecting events using multiple kinds of properties. Finally, it will discuss the selection of  $\eta$ .

### 8.1. Complexity and Performance Analysis of the Algorithm

**Theorem 2.** *The CETC algorithm would end within a limited time.*

*Proof.* From CETC algorithm, we can know that the algorithm is time synchronization. The algorithm has eight time periods corresponding to different calculation and operations of data sending. As shown in Figure 3, scale above marks data packets to be sent, and scale below is the corresponding time period. If exceeding the corresponding time period, the node will not deal with the corresponding data packets. Therefore, CETC algorithm will end within  $T_D + T_N + T_P + T_S + T_\tau + T_\tau/L + T_Q + T_{SS}$ .  $\square$

**Theorem 3.** *The total time complexity of CECT algorithm is  $O(\Delta' + \Delta \log(\Delta))$ , where  $\Delta$  indicates the maximum value of the number of nodes in the scope  $1/\eta \max\{R_{si}\}$  ( $i = 1, 2, \dots, m$ ) of network node and  $\Delta'$  shows the maximum number of nodes in the decision range of any node in the network, that is, the maximum node degree in figure  $G_D$ .*

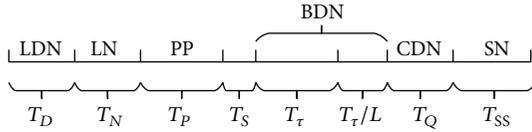


FIGURE 3: Time periods of CETC algorithm execution.

*Proof.* In Figure 3, the corresponding calculation operations must be carried out at each time period. In the time period  $T_D$ , the time complexity of broadcasting LDN packet of each node is  $O(1)$ . In the time period  $T_N$ , node needs to decide whether it can be a  $k$ -watching decision node; therefore, nodes need to deal with the received LDN packets one by one in the time period  $T_D$  and the time complexity is  $O(\Delta)$ . In the time period  $T_P$ , nodes need to calculate their own priorities and the time complexity is  $O(1)$ . In the time period  $T_S$ , nodes need to calculate the time of sending BDN data packet, so they need to find the maximum value and minimum value of priority level of the received PP data packet, and the time complexity is  $O(\Delta')$ . BDN data packets will be sent and received in the time period  $T_\tau + T_\tau/L$ ; the time complexity is  $O(1)$ . In the time period  $T_Q$ , non- $k$ -watching decision node that did not receive BDN data packet needs to look for the maximum priority level of the received PP data packet; the time complexity is  $O(\Delta')$ . In the time period  $T_{SS}$ , local decision node needs to extract the information of the node from LDN data packet and sorts nodes according to the energy of the node; then choosing the node with larger energy as sensor node disposable, the time complexity of sorting is  $O(\Delta \log(\Delta))$ . So CETC algorithm's total time complexity is  $O(\Delta' + \Delta \log(\Delta))$ .  $\square$

**Theorem 4.** *Message exchange complexity (Message exchange complexity) of CETC algorithm is  $O(N)$  and  $N$  is the total number of nodes in a network.*

*Proof.* In addition to exchanging message with specific node in the  $T_Q$  and  $T_{SS}$  period, nodes carry out broadcasting operations in the other period, so message exchange complexity of each node is  $O(1)$ . So, the message complexity of the entire network is  $O(N)$ .  $\square$

**Theorem 5.** *If nodes in the network are all  $k$ -watching decision nodes, when the CETC algorithm ends, the probability of two local decision nodes' distance less than  $R_d$  is small.*

*Proof.* If nodes in the network are all  $k$ -watching decision nodes, this means that each node can send BDN package. In the time period  $T_\tau + T_\tau/L$ , all the nodes could be controlled by the local decision node. Assume that there are two nodes  $u$  and  $v$ , whose distance is not larger than  $R_d$ , the waiting time before  $u$  sends BDN data packet is as follows:

$$t_u = \left( 1 - \frac{[(\text{Prio}_u - \text{Prio}_{\min}) / (\text{Prio}_{\max} - \text{Prio}_{\min})] \times L}{L} \right) \times T_\tau + T_{\text{random}}. \quad (21)$$

As seen from the above equation,  $T_\tau$  is divided into  $L$  segments; each node selects a time period according to the priority firstly and then chooses a random time  $T_{\text{random}}$  within this time period,  $T_{\text{random}} \in [0, T_\tau/L]$ . Because of the uncertainty in figure  $G_D$  and residual energy of  $u$  and  $v$ , and the priority of Prio is uncertain,  $u$  and  $v$  will choose a moment randomly in time period  $T_\tau + T_\tau/L$ . Assume that there is  $K$  moments within  $T_\tau/L$ . So the probability of node sending BDN data packets at any moment is as follows:

$$P_u = P_v = \frac{1}{K(L+1)}. \quad (22)$$

If  $u$  and  $v$  are local decision nodes, then  $u$  and  $v$  will send BDN data packets at the same time; the probability is

$$P = P_u \times P_v = \frac{1}{K^2(L+1)^2}. \quad (23)$$

When  $K = 10$  and  $L = 10$ ,  $P \approx 8.3 \times 10^{-5}$ . In fact, the value of  $K$  is bigger and the usage of collision detection algorithm would also be considered; as a result, the probability of the distance of two local decision nodes less than  $R_d$  is small.  $\square$

Theorem 5 shows that local decision node is geographically distributed when CECT algorithm ends; thus, it helps to balance energy consumption of the network.

**Theorem 6.** *When voids are not detected,  $V_D$  is a dominating set of graph  $G = (V, E)$ .*

*Proof.* The CETC algorithm shows that if the node sends BDN packets successfully, then STATE of the node is set to DECIDE. If  $u$  does not send or receive any BDN data packets, then  $u$  is not available  $k$ -watching decision node; the node will look for node  $v$  with biggest Prio within the scope of  $R_d$  as a local decision node. When voids are not detected,  $u$  always can find such a node  $v$ . So for any node in the network, a local decision node always can be found within the scope of  $R_d$ . It can be seen that  $V_D$  is a dominating set of graph  $G_D = (V, E_D)$ . Due to  $E_D \subseteq E$ ,  $V_D$  is a dominating set of graph  $G = (V, E)$ .  $\square$

From Theorem 6, we know that when the voids are not detected, the decision range of local decision node covers all nodes within the monitoring area, so nodes in the  $V_D + V_S$  are active; then they can monitor all Composite events in the position of the nodes, so as to monitor the occurrence situation of Composite event in the whole monitoring area approximately.

**Theorem 7.** *The set constituted of all active nodes is connected dominating set of  $G' = (V', E')$ .*

*Proof.* The DS algorithm shows that the set constituted of all active nodes is  $V_D \cup V_S \cup V_C \cup \{\sin k\}$ . In the network, any local decision node  $u$  sends activation message to its parent nodes  $u_{p1}, u_{p2}$ , and the activation message will be transmitted to the base station. Nodes on the Path  $u \rightarrow \sin k$  could only be local decision nodes, sensing nodes, or connection nodes; connection nodes are only possible on such path.

Accordingly, in the case that graph  $G' = (V', E')$  is connected, there is at least one path between local decision nodes and base station. Since  $V_D$  is the dominating set of graph  $G = (V, E)$ , so the  $V_D \cup \{\sin k\}$  is the dominating set of graph  $G' = (V', E')$ . Because  $V_S$  and  $V_C$  ensures the connectivity from nodes in  $V_D$  to the base station, so all the active nodes  $V_D \cup V_S \cup V_C \cup \{\sin k\}$  are the connected dominating set of graph  $G' = (V', E')$ .  $\square$

Theorem 7 shows that the connection node ensures the connectivity between local decision nodes and base stations, so the information of Composite event of the local decision node can be transmitted to the base station; if nodes in  $V_D \cup V_S \cup V_C$  are active, then they can complete the detection of events within the monitoring area and send the result of the local decision to the base station.

### 8.2. Analysis of Composite Events

**Theorem 8.** *Testing events with multiple attributes are more accurate than with one attribute.*

*Proof.* Assuming that Composite event is  $E$ , if using  $M$  properties can completely test  $E$ . Now there are only  $m \leq M$  kinds of sensors in the network; the judgment of the  $i$ th property is  $e_i$ .  $e_i$  ( $1 \leq i \leq m$ ) are not related; then  $\{E = 1\} = \{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\}$ .

Because  $\{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\} \subseteq \{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\}$ , then

$$\{E = 1\} \subseteq \{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\}. \quad (24)$$

When  $e_i = 1$ , the occurrence probability of composite event probability of  $\{E = 1\}$  is as follows:

$$\begin{aligned} P(\{E = 1\} | \{e_i = 1\}) &= \frac{P(\{E = 1\} \cap \{e_i = 1\})}{P(\{e_i = 1\})} \\ &= \frac{P(\{E = 1\})}{P(\{e_i = 1\})}. \end{aligned} \quad (25)$$

When testing various attributes, when  $e_1 = 1, e_2 = 1, \dots, e_m = 1$ , the probability of the Composite event is as follows:

$$\begin{aligned} P(\{E = 1\} | \{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\}) \\ &= \frac{P(\{E = 1\} \cap \{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\})}{P(\{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\})} \\ &= \frac{P(\{E = 1\})}{\prod_{i=1}^m P(\{e_i = 1\})}. \end{aligned} \quad (26)$$

In order to simplify the analysis,  $P(\{e_i = 1\}) = p_e$  ( $1 \leq i \leq M$ ); then

$$\begin{aligned} \kappa &= \frac{P(\{E = 1\} | \{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\})}{P(\{E = 1\} | \{e_i = 1\})} \\ &= \frac{1}{p_e^{m-1}} = \left(\frac{1}{p_e}\right)^{m-1}. \end{aligned} \quad (27)$$

If  $P_e$  is identified,  $1/P_e > 1$ ,  $k$  grows exponentially along with  $m$ . Therefore, testing the event with multiple attributes is more accurate than with one attribute.

In the ideal state, no event occurs,  $e_i = 0$ ; when the event occurs,  $e_i = 1$ . In real environment, the situation that  $\{e_i = 1 | H_0\}$  and  $\{e_i = 1 | H_1\}$  tends to appear, which is called that the node has been influenced. There may be two situations of the interference: the first is the failure of the node itself; the second is the interference caused by other factors in the environment. For example, in forest fire detection, temperature properties are  $T$ ; when  $\{T = 1 | \text{No Fire}\}$ , a node may fail, or the sensor is exposed to direct sunlight in summer, causing the perceived temperature data to be higher.  $E$  is the real value whether events happen in the environment;  $R$  is the determination result that according to  $m$  types of properties values of the event of the node,  $\{R = 1\} = \{e_1 = 1\} \cap \{e_2 = 1\} \cap \dots \cap \{e_m = 1\}$ .

Setting the probability that the node is interfered as  $p$ ,  $p = P(\{e_i = 1\} | H_0) = P(\{e_i = 0\} | H_1)$ , and the probability of the nodes correctly detecting the environment is as follows:

$$\begin{aligned} P_d &= P(\{R = 1\} | H_1) \times P(H_1) + P(\{R = 0\} | H_0) \times P(H_0) \\ &= (1 - p)^m p_E + (1 - p^m)(1 - p_E). \end{aligned} \quad (28)$$

The probability that node cannot detect the environment correctly is as follows:

$$\begin{aligned} P_f &= P(\{R = 1\} | H_0) \times P(H_0) + P(\{R = 0\} | H_1) \times P(H_1) \\ &= (1 - (1 - p)^m) p_E + p^m(1 - p_E). \end{aligned} \quad (29)$$

The  $P_E$  indicates the probability that event occurs. For given  $p$  and  $P_E$ , the influence that  $m$  has on  $P_d$  is analysed. Making derivation for  $m$ , when the derivative is zero, get  $m$  when  $P_d$  is the maximum value:

$$m = \frac{\ln(((1 - p_E) \ln p) / (p_E \ln(1 - p)))}{\ln(1 - p) - \ln(p)}. \quad (30)$$

$P_d$  first increases and then decreases along with  $m$ . Figure 4 is the situation of  $P_d$  when  $p = 0.1, P_E = 0$ .

If the detection is required with higher accuracy,  $m$  can take larger integers around the extreme value  $P_d$ . But for the important event, it is considered that the node can detect the change of attributes of the environment and has believable probability of determination result when the events occur; that is,

$$\begin{aligned} P_{Ed} &= P(\{E = 1\} | \{R = 1\}) \times P(\{R = 1\} | H_1) \times P(H_1) \\ &= \frac{P_E}{p_e^m} \times (1 - p)^m p_E = p_E^2 \left(\frac{1 - p}{p_e}\right)^m. \end{aligned} \quad (31)$$

So if testing important events, when  $(1 - p)/P_e < 1$ ,  $m$  should be as small as possible, but in order to ensure that  $P_d$  is larger,  $m$  should take larger integers around the extreme value  $P_d$ .

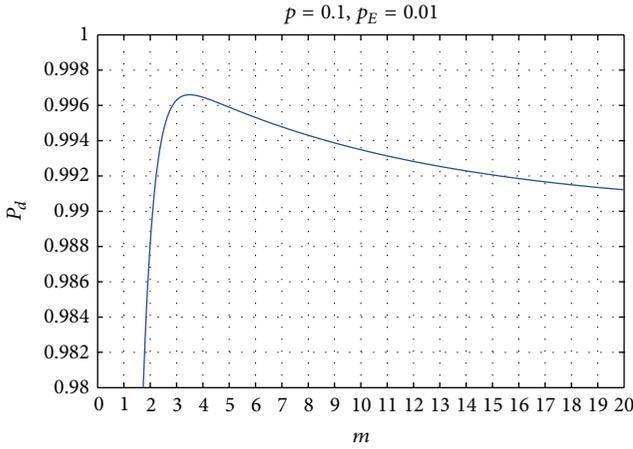


FIGURE 4: The relationship between  $P_d$  with  $m$ .

When  $(1-p)/P_e \geq 1$ ,  $m$  is the large the better. When  $m \rightarrow \infty$ ,  $\lim_{m \rightarrow \infty} P_d = 1 - p_E$ . Since  $P_d$  is monotonically decreasing, so when  $m$  takes a larger value, the probability of node accurately testing the environment is still great as long as  $P_E$  is lesser.  $\square$

8.3. Selection of  $\eta$ . According to the CETC algorithm,  $R_C$  must meet the conditions that  $R_C \geq R_d = (1 - \eta) \min\{R_{S_i}\}$  and  $R_C \geq \max\{\eta R_{S_i}\}$  ( $1 \leq i \leq m$ ); then

$$R_C \geq \max\left(\left(1 - \eta\right) \min\left\{R_{S_i}\right\}, \eta \max\left\{R_{S_i}\right\}\right) \quad (1 \leq i \leq m). \quad (32)$$

To simplify the analysis, assume that there are  $m$  kinds of nodes deployed in area  $A$  randomly, the  $i$ th node integrated sensor is  $C_i$  and the number of this kind of nodes is  $N_i$ . For the node  $u$  integrated  $C_i$ ,  $B_i$  denotes that there is the  $i$ th node in the range  $R_{S_i}/\eta$  ( $\eta$  is the integer greater than 1) from  $u$ , and  $D$  indicates that  $u$  can be the local decision node; then

$$P(B_i) = 1 - \prod_{i=1}^{N_i} \frac{A - \pi(R_{S_i}/\eta)^2}{A} = 1 - \left(\frac{A\eta^2 - \pi R_{S_i}^2}{A\eta^2}\right)^{N_i}. \quad (33)$$

$$P(D) = \left(\prod_{i=1, i \neq \ell}^m P(B_i)^k\right) \times P(B_\ell)^{k-1} \quad (34)$$

When deploying nodes, we require that  $P(D) \geq \theta$ , where  $\theta$  is the given threshold. For  $m, N_i, R_{S_i}$  ( $1 \leq i \leq m$ );  $P(D)$  will decrease along with  $n$  increasing. When  $P(D) = \theta$ , we will get a biggest integer  $\eta$ , and it is the optimal solution. So  $\eta$  makes the decision range biggest and decreases the number of local decision nodes and active nodes. Assuming that  $N_i = N_0, R_{S_i} = R_{S_0}, 1 \leq i \leq m$ , and  $k = 1$ , then

$$\begin{aligned} P(B_i) &= 1 - \prod_{i=1}^{N_i} \frac{A - \pi(R_{S_0}/\eta)^2}{A} \\ &= 1 - \left(1 - \frac{\pi R_{S_0}^2}{A\eta^2}\right)^{N_0} \end{aligned}$$

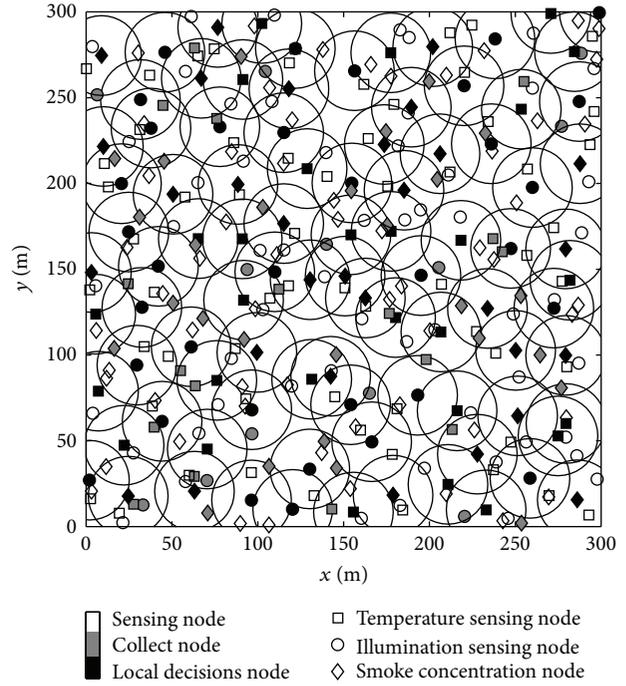


FIGURE 5:  $N = 300$ .

$$\begin{aligned} P(D) &= \left(1 - \left(1 - \frac{\pi R_{S_0}^2}{A\eta^2}\right)^{N_0}\right)^{m-1} \geq \theta \\ \eta &= \left\lceil \sqrt{\frac{\pi R_{S_0}^2}{A} \left(1 - (1 - \theta^{1/(m-1)})^{1/N_0}\right)^{-1}} \right\rceil. \end{aligned} \quad (35)$$

### 9. Experiment and Discussion

We do the simulation experiments and analysis about the accuracy, error rate, energy consumption, the fault detection ability, and decision mechanism of the proposed algorithm in Java and MATLAB. We perform a fire detecting in area  $300 \times 300 \text{ m}^2$ ; three kinds of sensor nodes are deployed randomly in the area to perceive the temperature, light intensity, and smoke concentration, respectively, the initial energy of the node is 1J, and the size of packet is 128 bits. The number of each kind of nodes is  $N_0$ ; the largest communication radius of the sensor is 40 m, taking  $\theta = 90\%$ ,  $k = 1$ . Next we will take simulation experiment under the condition  $N_0 = 300, 400, \dots, 1300$ , the corresponding optimal  $\eta = 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4$ . The area of Composite event is a circular area with radius 100 m; the Composite event happens randomly. The experimental data is the average of 50 times simulation.

Figure 5 is the condition when  $N_0 = 300$ , there is a total of 113 local decision nodes, and sleeping nodes are not marked. Distribution of decision node is more scattered. There still exist a few areas that have not been covered, though they almost cover the entire monitoring area. We only need to

TABLE 2: Settings of simulation parameter.

Parameter	Value
The area of $A$	$300 \times 300 \text{ m}^2$
The maximum radius of $R_c$	40 m
Total number of nodes, $N$	300–13000
Initial energy	1 J
$E_{\text{elec}}$ ( $\text{nJ} \cdot \text{bit}^{-1}$ )	50
The size of data package	128 bits
The size of control package	12 bits

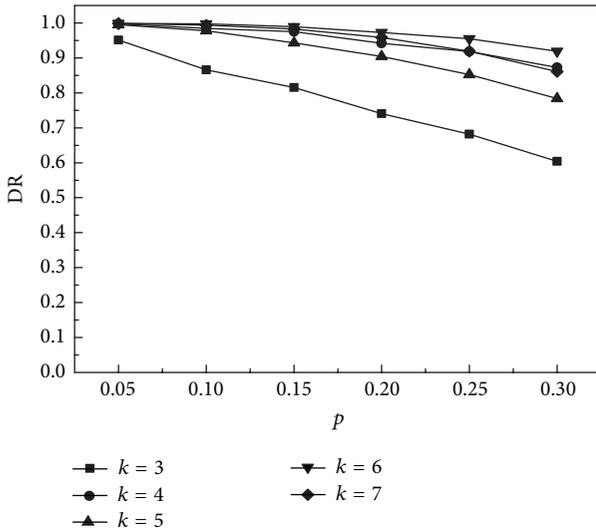


FIGURE 6: The relationship of DR with  $p$ .

determinate these nodes to make sure whether there is a fire in the monitoring area. The experimental parameters are shown in Table 2.

**9.1. Failure Detection.** Assume that the probability of node failure is  $p$ . For the faulty node, it will produce the wrong perception attribute values. We separate the wrong data from the sensing data by the two-dimensional  $\tau$ -GAS algorithm. Next we will analyze the relationship between  $K$  and detection accuracy and the false alarm rate to shown their influence on the two-dimensional  $\tau$ -GAS algorithm. Precision (DR) is defined as the ratio of the number of the fault nodes detected accurately and the number of actual fault nodes in the network. The false alarm rate (FAR) is the ratio of the number of nonfaulty nodes which are wrongly detected and the number of normal nodes in the network.

From Figure 6, when  $k$  is the same, the larger  $p$  is and the smaller DR is; that means under the same conditions, detection accuracy will be reduced when there are much fault nodes. When  $p$  is the same, we found that DR is bigger when  $k$  is even; this is because such an assumption is made in the simulation that the node can always choose the right category to make fuzzy decision-making when the number of two types of data is the same. From the overall trend, DR increases as  $k$  increases, and this is because the larger the  $k$  is, the more

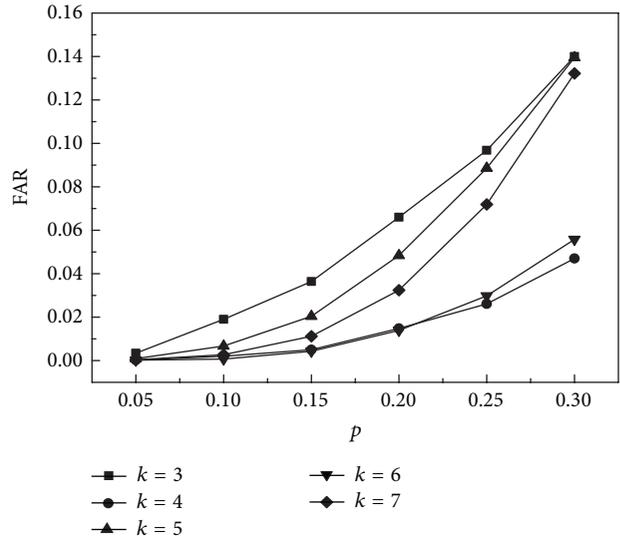


FIGURE 7: FAR's relationship with  $p$ .

the nodes are that will provide the attribute information of the environment; then the detection accuracy will be improved.

As can be seen from Figure 7, when  $k$  is even, the above problem still exists. From the overall trend, FAR decreases along with the increase of  $k$ . This is because that the greater the  $k$  is, the more the nodes are that provide the attribute information of the environment, and the error rate will reduce.

By formula (34), when given  $\eta$  and the number of nodes in the network,  $P(D)$  monotonically decreases along with the increase of  $k$ . Since  $P(D) \geq \theta$ , the optimal solution of  $k$  is the max value of  $k$  when  $P(D) \geq \theta$ . In practical applications, the  $\eta$  and  $k$  may all be variable, larger  $\eta$  helps to reduce the energy consumption in network; a larger  $k$  could improve the accuracy of event detection and fault detection. But when given the number of nodes,  $P(D)$  will decrease rapidly along with the increase of  $\eta$  and  $k$ . Therefore, in practice, we should establish a trade-off between energy consumption and detection accuracy and choose reasonable  $\eta$  and  $k$ .

**9.2. Energy Consumption and Accuracy.** This experiment evaluates the energy consumption situation of the detection algorithm through the number of active nodes and the exchange capacity of the information when event occurs. Active node refers to the node which is in a nondormant state; capacity of message exchange refers to the number of all the packets which are received by the nodes in the network and used to detect the event.

Figure 8 reflects the number of active nodes of the three algorithms. There are more active nodes in COLLECT algorithm because of lack of sleep scheduling. FL-CED and CDS algorithms use sleep scheduling mechanism so that the number of active nodes reduced greatly. From the figure, the larger the  $\eta$  in FL-CED algorithm is, the smaller the number of active nodes is; this is due to the fact that the greater  $\eta$  leads to larger decision range of the node, so the monitoring

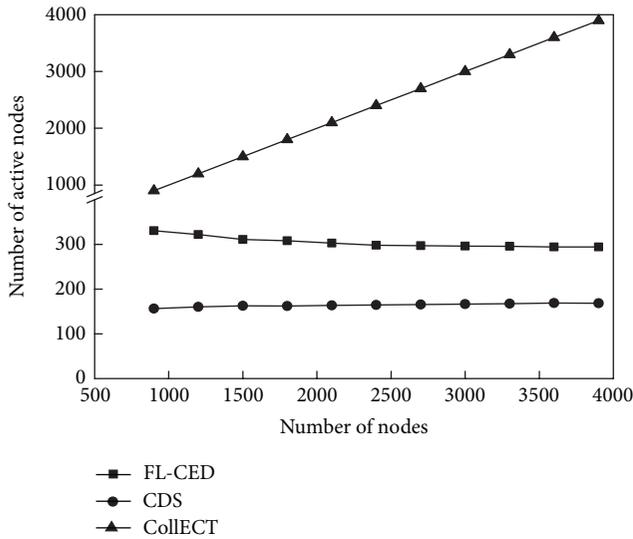


FIGURE 8: Number of active nodes.

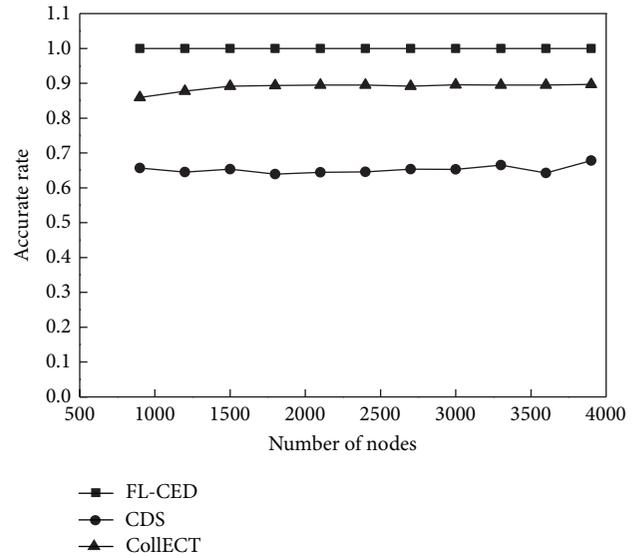


FIGURE 10: Accurate rate.

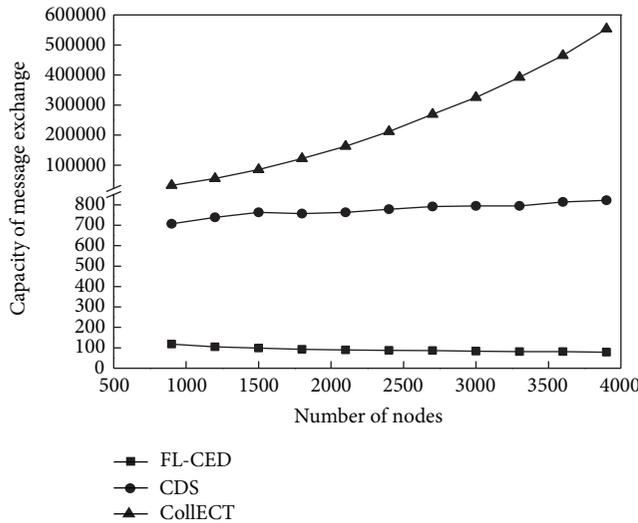


FIGURE 9: Capacity of message exchange.

area could be tested with fewer active nodes in the network. Since the perception range of the node is not considered in CDS algorithm, the node just sends perceived environmental attributes to communication neighbors, so the number of activation nodes is fewer compared to FL-CED algorithm, but we will see that the performance of the FL-CED algorithm is better than the CDS algorithm in the detection accuracy and message exchange capacity.

Figure 9 reflects the comparison of message exchange capacity of three algorithms. The figure shows that COLLECT algorithm requires a large amount of data exchange when event occurs. Sensor nodes in FL-CED algorithm only need to send the perceived environmental attributes to the selected local decision node; thus, this one-way transmission mechanism reduces the message exchange capacity further compared to the way of broadcasting adopted in CDS algorithm.

Document [35] defines the accurate rate and out-event false rate to assess the accuracy of the algorithm. Set  $S_{evt}$  as the set of active nodes in the area of the event, in FL-CED algorithm,  $S_{evt}$  is the set of the local decision node in the area of the event,  $S_{urg}$  is a collection of nodes that indicates an event occurs, and  $N$  is the total number of nodes.

*Accurate Rate.* Accurate rate is the ratio of the number of nodes that indicates an event occurs and the total number of nodes in the area of event, noted as  $\mathcal{A}(S_{urg})$ ,  $\mathcal{A}(S_{urg}) = |S_{urg} \cap S_{evt}|/|S_{evt}|$ . Out - the Event false rate: the ratio of the number of nodes indicating an event occurs out of the event area and the total number of nodes out of the event area, noted as  $\mathcal{E}_o(S_{urg})$ ,  $\mathcal{E}_o(S_{urg}) = (|S_{urg}| - |S_{urg} \cap S_{evt}|)/(N - |S_{evt}|)$ .

Figure 10 reflects accuracy of three algorithms; from the figure, the accuracy of FL-CED algorithm is close to 100%; this is due to that the perception node in the event region is able to correctly perceive events and send it to local decision node for judging. In COLLECT algorithm, it is not necessarily that nodes perceiving all properties of the Composite event in network topology constituted of the node and its neighbor nodes exist; therefore, judgment that nodes make has mistakes. The accuracy of CDS algorithm is not as good as FL-CED algorithm and COLLECT algorithm. Figure 11 reflects the out-event false rate of the node: the out-event false rate of CDS algorithm is low, there is little difference between the out-event false rate of FL-CED algorithm and COLLECT, around 35%, the reason why the out-event false rate of the event reaches to 35% is that the area of events is larger in the experiment, and the monitor area is smaller, but this does not affect the performance comparison of the algorithm. Although the accuracy of CDS algorithm is lower, the probability of misjudging the event of the node which is in event peripheral is lower, so CDS algorithm can reflect the external events more accurately. Because we tend to attach importance to determine the situation within the event area,

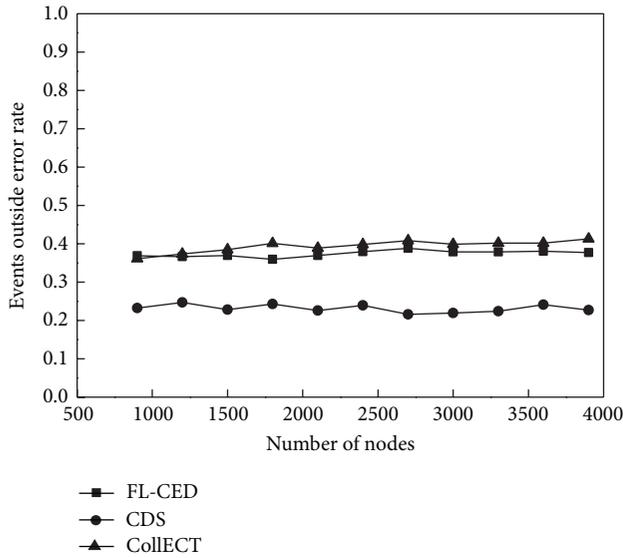


FIGURE 11: Events outside the error rate.

so overall, accuracy of FL-CED algorithm is superior to COLLECT algorithm and CDS.

Since different incident detection algorithms may use different routing algorithms, so the performance of life cycle is not possible to be compared. However, from the number of active nodes and message exchange capacity, we can know that the energy consumption of FL-CED algorithm is less compared with the other two algorithms. In addition to energy consumption, energy balance is also an indicator affecting the network life cycle. Assuming that the incident happens once each round, the area is round with radius 100 m; the Figure 12 shows energy distribution after FL-CED algorithm having performed 250 rounds when  $N_0 = 300$ ; it can be seen from the diagram that energy consumption in the network is more balanced.

**9.3. Fuzzy Decision of Composite Event.** Data used in this study is fire testing data [35] released from the USA NIST. We will test the fire with temperature  $V_1$  and smoke concentration  $V_2$ . Solution set is  $\{V_1, V_2, V_3\}$ , where  $V_1$  notes sending fire event data,  $V_2$  notes sending warning data, indicating that environment is unusual, and  $V_3$  notes that no event occurs and does nothing. Constructing membership functions is based on given thresholds [37] of smoke sensor and thermal sensors Weighting matrix is  $w = \{0.5, 0.5\}$ .

In Figures 13(a), 13(b), and 13(c) are membership functions  $u_{11}(x)$  and  $u_{21}(x)$  and  $u_{31}(x)$  related to temperature, respectively. In Figures 13(d), 13(e), and 13(f) are the membership functions  $u_{12}(x)$ ,  $u_{22}(x)$  and  $u_{32}(x)$  are related to smoke concentration, respectively. Figure 14 is the membership of three schemes, and the origin is the burning time. Before catching fire,  $u_3$  is larger, so the nodes use the scheme  $A_3$ , and they judge that there is no fire. At the beginning of fire, the sensor does not detect enough intensity to determine events, so  $u_3$  is larger. After a period of time, the fire expands, so the node can percept it, there is a dramatic increase in  $u_1$ ,  $u_3$

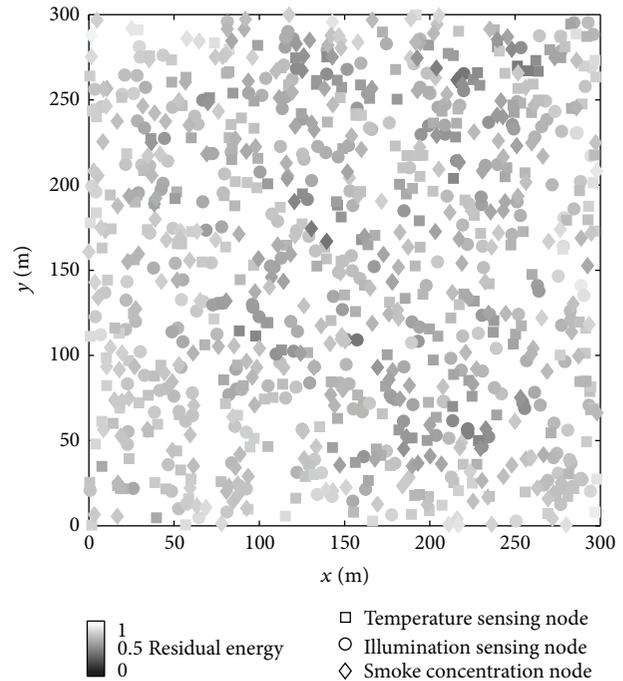


FIGURE 12: Energy distribution of DS algorithm after 250 rounds.

sharply reduce, and  $u_2$  rises a little;  $u_1$  is the largest, and the nodes take scheme  $A_1$ ; they judge that the fire happened. It can be seen from Figure 14 that nodes can effectively perceive the happening of the fire.

### 10. The Conclusion

Composite event is a kind of event which widely exists in the nature; the event-driven wireless sensor network is mainly used for detecting the Composite event, such as fire and intrusion detection. Due to the limitation of node itself, nodes cannot integrate all types of sensors required to detect composite events. However, the detection of Composite event requires collaboration of multiple nodes to complete. By analyzing the characteristics of the composite event, considering that different sensors have different perception, the paper has proposed a criterion of Composite event detection and designed the network topology structure algorithm CETC based on the dominating set; the algorithm will choose part of nodes in the network as local decision nodes and select a specific node as the sensing node. Local decision node makes decisions on the Composite event within its decision scope. Two-dimensional  $\tau$ -GAS algorithm being able to screen test data correctly and decision making mechanism based on fuzzy decision are proposed. Two-dimensional  $\tau$ -GAS algorithm can effectively filter the errors data perceived by failed node through the sliding window mechanism and clustering analysis and then calculate an estimated value of the current environment attributes. Node can determine which sensors have a permanent fault through calculating the probability of failure of the sensor. For the ambiguity of the definition of the composite event, this paper puts forward a

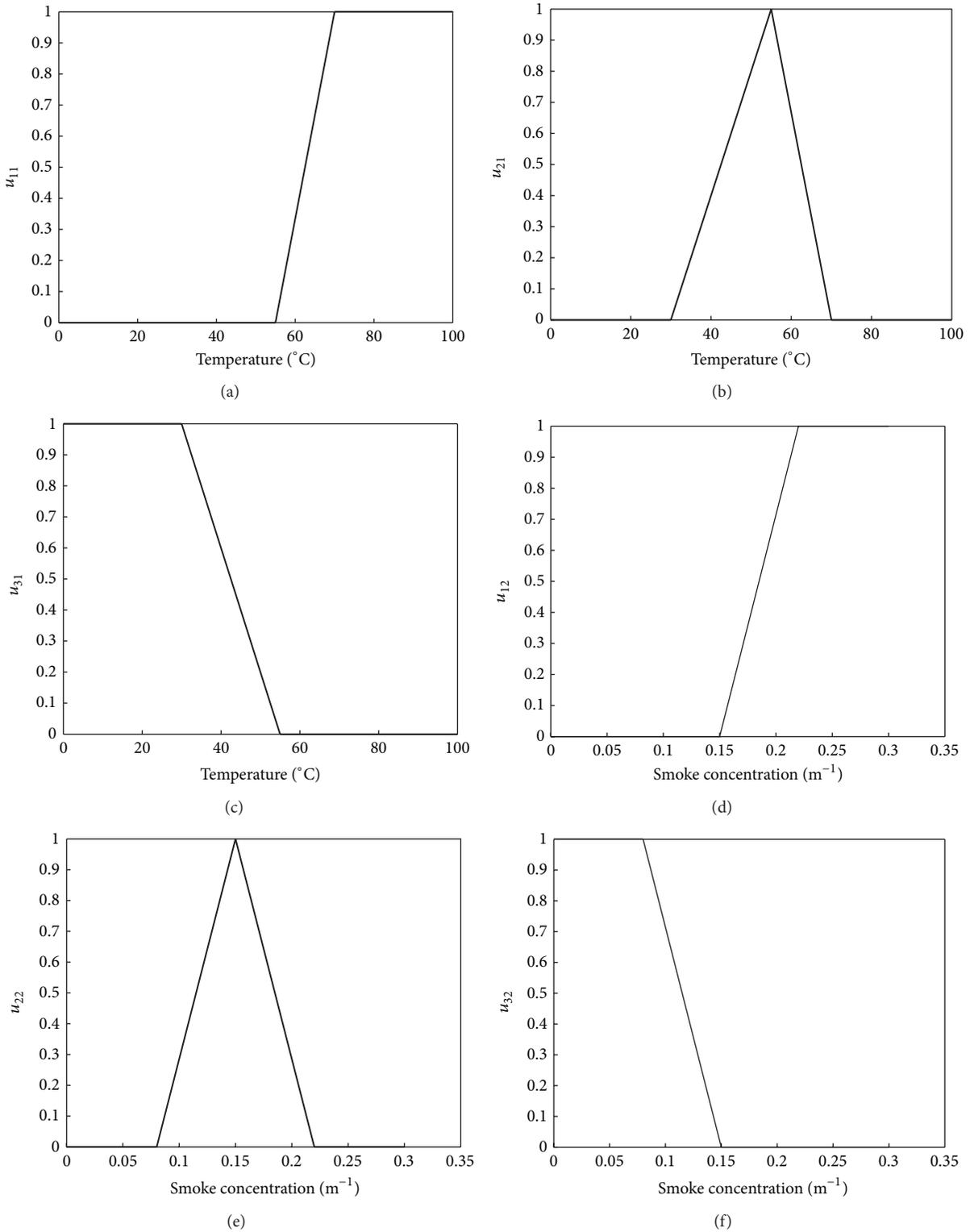


FIGURE 13: Membership functions of attributes corresponding to scheme.

decision mechanism based on fuzzy decision; this decision mechanism can determine whether the event has happened effectively. Compared with the decision mechanism based on fuzzy logic in the paper, this decision mechanism does

not need a huge rule base. It retains the advantages of the decision-making mechanism based on fuzzy logic and can adjust to the event detection in different environment by configuring membership function matrix  $I$  and weight matrix  $w$ .

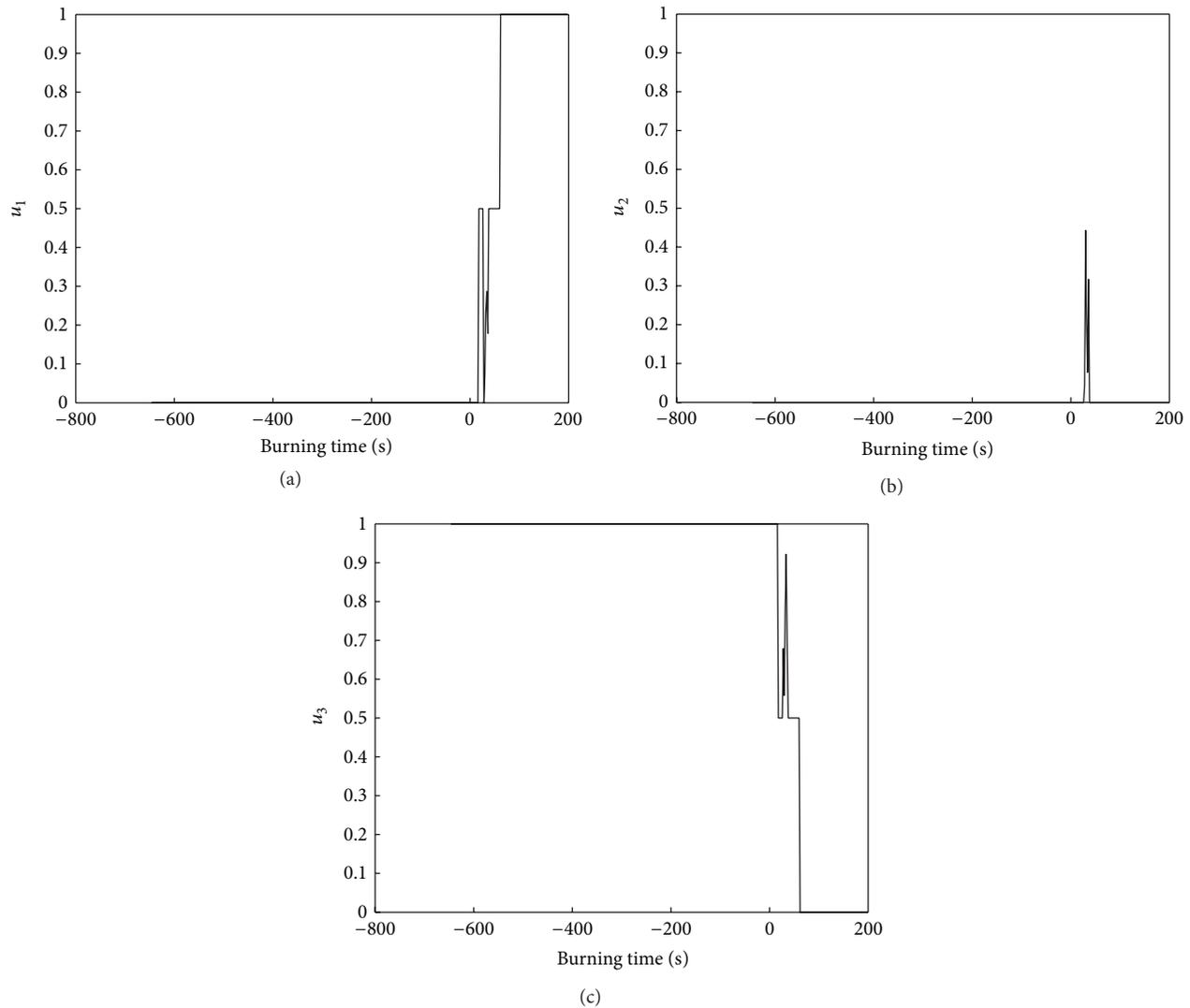


FIGURE 14: Program membership.

The Composite event in this paper is static; it is not suitable for detecting dynamic Composite event in motion. Therefore, the tracking for mobile composite event still needs further research.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants nos. 61070169 and 61201212, Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, Production Teaching Research Prospective of Jiangsu Province no. BY2012114, Suzhou Key Laboratory of Converged Communication nos. SKLCC2013XX and

SZS0805, and Application Foundation Research of Suzhou of China no. SYG201239.

### References

- [1] H. X. Tan, M. C. Chan, W. D. Xiao, P.-Y. Kong, and C.-K. Tham, "Information quality aware routing in event-driven sensor networks," in *Proceedings of the IEEE INFOCOM*, pp. 1-9, San Diego, Calif, USA, March 2010.
- [2] C. T. Vu, R. A. Beyah, and Y. Li, "Composite event detection in wireless sensor networks," in *Proceedings of the 27th IEEE International Performance Computing and Communications Conference (IPCCC '07)*, pp. 264-271, Atlanta, Ga, USA, April 2007.
- [3] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241-250, 2004.

- [4] K. K. Chintalapudi and R. Govindan, "Localized edge detection in sensor fields," in *Proceedings of Sensor Network Protocols and Applications*, pp. 59–70, Los Angeles, Calif, USA, 2003.
- [5] V. Guo, Q. Chen, and P. Fan, "On the fault-tolerant event region detection scheme with user specified threshold in wireless sensor networks," in *2009 International Conference on Communications, Circuits and Systems (ICCCAS '09)*, pp. 196–201, Chengdu, China, July 2009.
- [6] Y. B. Wang, M. C. Vuran, and S. Goddard, "Analysis of event detection delay in wireless sensor network," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 1296–1304, Lincoln, Neb, USA, 2011.
- [7] A. V. U. P. Kumar, V. A. M. Reddy, and D. Janakiram, "Distributed collaboration for event detection in wireless sensor networks," in *Proceedings of the 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC '05)*, pp. 1–8, New York, NY, USA, 2005.
- [8] K. P. Shih, S. S. Wang, P. H. Yang, and C. Chang, "COLLECT: collaborative event detection and tracking in wireless heterogeneous sensor networks," in *Proceeding of the 11th IEEE Symposium on Computers and Communications (ISCC '06)*, pp. 935–940, Taiwan, June 2006.
- [9] G. Amato, S. Chessa, C. Gennaro, and C. Vairo, "Efficient detection of composite events in wireless sensor networks: design and evaluation," in *Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC '11)*, pp. 821–823, Kerkyra, Greece, July 2011.
- [10] Y. S. Li, C. Y. Ai, C. T. Vu, Y. Pan, and R. Beyah, "Delay-bounded and energy-efficient composite event monitoring in heterogeneous wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1373–1385, 2010.
- [11] S. Lai, J. Cao, and X. Fan, "TED: efficient type-based composite event detection for wireless sensor network," in *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '11)*, pp. 1–8, Kowloon, China, June 2011.
- [12] Y. Yang, A. Ambrose, and M. Cardei, "Coverage for composite event detection in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 8, pp. 1168–1181, 2011.
- [13] K. Shih, S. Wang, H. Chen, and P. Yang, "COLLECT: collaborative event detection and tracking in wireless heterogeneous sensor networks," *Computer Communications*, vol. 31, no. 14, pp. 3124–3136, 2008.
- [14] M. Marta, Y. Y. Yang, and M. Cardei, "Energy-efficient composite event detection in wireless sensor networks," in *4th International Conference on Wireless Algorithms, Systems, and Applications (WASA '09)*, pp. 94–103, Boston, Mass, USA, 2009.
- [15] B. Fu, R. Li, X. Xiao, C. Liu, and Q. Yang, "Non-interfering multipath geographic routing for wireless multimedia sensor networks," in *Proceedings of the 1st International Conference on Multimedia Information Networking and Security (MINES '09)*, pp. 254–258, Changsha, China, November 2009.
- [16] T. P. Nghiem and T. H. Cho, "A multi-path interleaved hop-by-hop en-route filtering scheme in wireless sensor networks," *Computer Communications*, vol. 33, no. 10, pp. 1202–1209, 2010.
- [17] T. A. Babbitt, C. Morrell, B. K. Szymanski, and J. W. Branch, "Self-selecting reliable paths for wireless sensor network routing," *Computer Communications*, vol. 31, no. 16, pp. 3799–3809, 2008.
- [18] I. Stojmenovic, D. Simplot-Ryl, and A. Nayak, "Toward scalable cut vertex and link detection with applications in wireless ad hoc networks," *IEEE Network*, vol. 25, no. 1, pp. 44–48, 2011.
- [19] K. Akkaya, F. Senel, and A. Thimmapuram, "Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility," *IEEE Transactions on Computers*, vol. 59, no. 2, pp. 258–271, 2010.
- [20] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks*, vol. 18, no. 4, pp. 319–349, 1988.
- [21] K. Kapitanova, S. H. Son, and K. Kang, "Using fuzzy logic for robust event detection in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 4, pp. 709–722, 2012.
- [22] I. Gupta, D. Riordan, and S. Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks," in *Proceedings of the 3th Annual Communication Networks and Services Research Conference (CNSR '05)*, pp. 255–260, Halifax, Canada, May 2005.
- [23] J. Kim, S. Park, Y. Han, and T. Chung, "CHEF: Cluster Head Election mechanism using Fuzzy logic in wireless sensor networks," in *Proceedings of the 10th International Conference on Advanced Communication Technology (ICACT '08)*, pp. 654–659, Gangwon-Do, Republic of Korea, February 2008.
- [24] H. Y. Lee and T. H. Cho, "Fuzzy logic based key disseminating in ubiquitous sensor networks," in *Proceedings of the 10th International Conference on Advanced Communication Technology*, pp. 958–962, Suwon, South Korea, February 2008.
- [25] H. B. Kim, H. Y. Lee, and T. H. Cho, "Fuzzy key dissemination limiting method for the dynamic filtering-based sensor networks," in *Proceedings of the 3rd International Conference on Intelligent Computing*, pp. 263–272, Qingdao, China, 2007.
- [26] B. Lazzarini, F. Marcelloni, M. Vecchio, S. Croce, and E. Monaldi, "A fuzzy approach to data aggregation to reduce power consumption in wireless sensor networks," in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS '06)*, pp. 457–462, IEEE, Montreal, Canada, June 2006.
- [27] M. J. Kim and T. H. Cho, "Routing path generation for reliable transmission in sensor networks using ga with fuzzy logic based fitness function," in *Proceedings of the International Conference on Computational Science and Its Applications*, pp. 637–648, Kuala Lumpur, Malaysia, 2008.
- [28] S. Y. Chiang and J. L. Wang, "Routing analysis using fuzzy logic systems in wireless sensor networks," in *Proceedings of the 21th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, pp. 966–973, Zagreb, Croatia, 2008.
- [29] Y. W. E. Chan and B. H. Soong, "A new lower bound on range-free localization algorithms in wireless sensor networks," *IEEE Communications Letters*, vol. 15, no. 1, pp. 16–18, 2011.
- [30] W. Chiu, B. Chen, and C. Yang, "Robust relative location estimation in wireless sensor networks with inexact position problems," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 935–946, 2012.
- [31] K. Ota, M. Dong, H. Zhu, S. Chang, and X. Shen, "Traffic information prediction in Urban Vehicular Networks: a correlation based approach," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '11)*, pp. 1021–1025, Cancún, Mexico, March 2011.

- [32] R. Tan, G. Xing, B. Liu, J. Wang, and X. Jia, "Exploiting data fusion to improve the coverage of wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 450–462, 2012.
- [33] C. Shuili, J. Li, and X. Wang, *Fuzzy Set Theory and Its Applications*, Science Press, Beijing, China, 2005.
- [34] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, San Diego, Calif, USA, 4th edition, 2009.
- [35] "Home Smoke Alarm Tests [EB/OL]," <http://smokealarm.nist.gov/>.
- [36] Z. Zhongcheng, C. Rao, and W. Cheng, "A compromise type fuzzy decision model and its application," *Strategy and Management*, vol. 14, no. 5, pp. 30–40, 2005.
- [37] J. A. Geiman and D. T. Gottuk, "Alarm thresholds for smoke detector modeling," in *Proceedings of the 7th International Symposium on Fire Safety Science*, vol. 7, pp. 197–208, June 2002.

