

Research Article

Boundary Region Detection for Continuous Objects in Wireless Sensor Networks

Yaqiang Zhang ¹, Zhenhua Wang ¹, Lin Meng,² and Zhangbing Zhou ^{1,3}

¹School of Information Engineering, China University of Geosciences, Beijing, China

²College of Science and Engineering, Ritsumeikan University, Kyoto, Japan

³Computer Science Department, Telecom SudParis, Évry, France

Correspondence should be addressed to Zhenhua Wang; wangzh@cugb.edu.cn

Received 26 January 2018; Accepted 1 April 2018; Published 9 May 2018

Academic Editor: Xuyun Zhang

Copyright © 2018 Yaqiang 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.

Industrial Internet of Things has been widely used to facilitate disaster monitoring applications, such as liquid leakage and toxic gas detection. Since disasters are usually harmful to the environment, detecting accurate boundary regions for continuous objects in an energy-efficient and timely fashion is a long-standing research challenge. This article proposes a novel mechanism for continuous object boundary region detection in a fog computing environment, where sensing holes may exist in the deployed network region. Leveraging sensory data that have been gathered, interpolation algorithms have been applied to estimate sensory data at certain geographical locations, in order to estimate a more accurate boundary line. To examine whether estimated sensory data reflect that fact, mobile sensors are adopted to traverse these locations for gathering their sensory data, and the boundary region is calibrated accordingly. Experimental evaluation shows that this technique can generate a precise object boundary region with certain time constraints, and the network lifetime can be prolonged significantly.

1. Introduction

The concept of fog computing has attracted more and more attention nowadays, since fog computing promises to provide a relatively low latency and high-efficiency service [1, 2]. Different from cloud computing, fog computing is geographically closer to the network edge, where the majority of heterogeneous Internet of Things (IoT) devices are located. Consequently, applications can completely or, at least partially, be achieved upon edge nodes, and, hence, sensory data may not be required to be routed to the cloud in order to reduce the network traffic. One of the typical applications for fog computing in industrial IoT is disaster tracking and management [3–5], where toxic liquid like vitriol would cause soil acidification and, even worse, toxic gases like ammonia gas would be harmful to human body once leaked. In industrial applications like factory monitoring, sensors are deployed to monitor gas, liquid, or other kinds of dangerous objects. In this article, we concentrate on continuous object boundary detection, where traditional techniques have explored this topic [6–8]. As shown in Figure 1, wireless sensor networks

(WSNs) serve as the foundation of IoT. In the context of fog computing, sensory data provided by sensor nodes are routed to contiguous fog nodes for further processing, and these sensory data are required to be routed to the cloud since fog nodes are weak in capacity to handle these data. Therefore, analyzing these data at the fog nodes in an efficient manner and, thus, detecting the boundary region of continuous objects in an accurate and (near) real-time manner are a promising research topic [9].

Energy efficiency is the key issue in IoT, since IoT smart things are mostly battery-powered, and they are hardly to be recharged due to the harsh working environment. Traditional techniques have been proposed, where an energy-efficient algorithm for boundary detection and monitoring (COBOM) is proposed for boundary detection by selecting a set of boundary nodes [10]. Each sensor node saves its neighbor nodes status by a bit array. The head sensor node suppresses sensory data of its neighbors and reports the aggregated data packet to the sink node. In a certain round of data reporting, a few representative boundary nodes would be selected to report their sensory data.

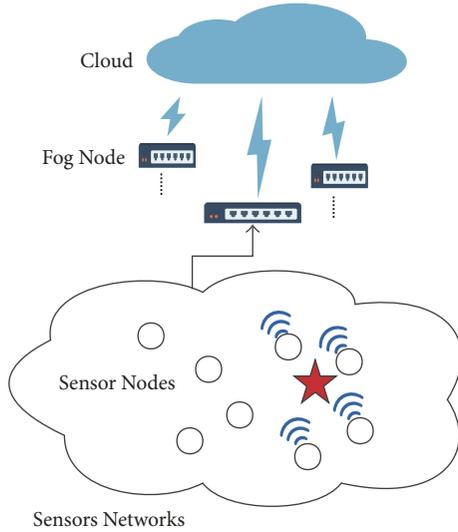


FIGURE 1: The network model in fog computing environment.

Sleep scheduling is applied to reduce energy consumption in sensor networks. In [11], the authors proposed a sleep scheduling scheme for object boundary monitoring. A few sensor nodes around the already existing boundary should be activated, while other nodes are stayed on the sleep mode. Generally, these schemes adopt partial of sensor nodes for data gathering purpose and, thus, for decreasing the network overhead. However, this strategy may lead to a coarse object boundary detection, since many sensor nodes may not report their sensory data to the sink to support the decision-making.

Research has been conducted on precise object boundary detection. In [12], the authors proposed a dynamic cluster structure for detecting and tracking a precise object boundary. Boundary nodes are dynamically organized into certain clusters. The head node in each cluster collects sensory data from other sensor nodes and sends these data to the sink node. More energy is consumed on data transmission so as to get a precise boundary. In [13], a data aggregation scheme is proposed to reduce the number of data packages among sensor nodes in the network. To detect and track a precise boundary, a piecewise quadratic polynomial interpolation scheme is applied. However, this technique does not check the predicted boundary as it may not be correct in some part of the predicted boundary.

To cover sensing holes and detect a precise boundary of continuous objects [14, 15], the authors proposed a mechanism through adopting mobile sensors for a precise and efficient boundary detection, where obstacles are not considered [16]. Generally, a few static sensor nodes are deployed, while mobile sensors are applied to detect sensing holes. In traditional schemes, object boundary shape is influenced by density and deployment of sensor nodes. This means that the more the sensor nodes are deployed, the more precise the boundary should be derived. Besides, the relatively large amount of sensory data should cause the network congestion. Note that a static sensor network could hardly generate a more precise object boundary when the object

keeps relatively stable, since there may be very few sensor nodes deployed in the boundary region. In this proposed scheme, mobile sensors traversing along an interpolation-based estimated boundary would detect a more precise object boundary.

This article proposes an efficient mechanism for object boundary detection in fog computing environment. Mobile sensors collaborate to bypass sensing holes for generating a precise object boundary, while the energy efficiency is a main concern. The main contributions of this proposed mechanism are summarized as follows:

- (i) Spatial interpolation algorithms are adopted to estimate the object boundary in the network region, where sensory data provided by static sensor nodes serve as the foundation. A threshold, which reflects the occurrence of potential events, is prescribed, and the possible object boundary is generated and represented as a curve. Potential points on the curve, which correspond to cruising stopping locations of mobile sensor nodes, are identified, in order to evaluate the applicability of the object boundary according to corresponding sensory data.
- (ii) An energy balanced mechanism is developed to schedule the itineraries of mobile sensors. Specifically, the time of cruising time of mobile nodes is estimated through calculating the time required to visit all points in the curve. The number of mobile sensors is derived when considering the time constraint of certain applications. The itinerary scheduling of multiple mobile sensors can be reduced to a multiobjective optimization problem, which can be solved through adopting heuristic algorithms.

Extensive evaluations are conducted to evaluate the accuracy of object boundary and the performance of mobile sensors deployment technique. The results show that the proposed mechanism for object boundary region detection can effectively find a precise object boundary. Meanwhile, the utilization of mobile sensors can discover sensing holes with certain time constraints.

This article is organized as follows. Section 2 introduces the concepts and related techniques. Section 3 presents the network initialization mechanism. The precise boundary area detection mechanism is developed in Section 4. Section 5 evaluates the technique developed in this article. Section 6 reviews and discusses relevant techniques, and Section 7 concludes this article.

2. Preliminary

This section introduces the network model and defines relevant concepts. Algorithms including spatial interpolation and planar graph are presented.

2.1. Networks Model. Sensor nodes are deployed in a two-dimensional network region. There are two types of sensing mode of sensor nodes. Fixed sensing range means that the sensing radius of sensor nodes cannot be changed, while adjusted sensing range means otherwise. In this article, we

adopt a fixed sensing range for sensor nodes. We define that the sensing range for both types of sensor nodes are nonadjustable. Different from [17], we give the probabilistic sensing model, and the probability of a sensor detecting an event at a distance x is defined as follows:

$$P(x) = \begin{cases} 1 & \text{if } x \leq r_s \\ 0 & \text{if } x > r_s, \end{cases} \quad (1)$$

where r_s represents the sensing radius of sensor nodes.

Generally, the communication radius of mobile nodes is larger than that of static sensors. The energy consumption of mobile nodes is mainly for their movement and communication process. To define a shape and a boundary for objects, a curve should be identified and corresponding to the threshold that defines the occurrence of a certain event.

Definition 1 (boundary line (BL)). It is a curve where the sensing value is equal to the prespecified threshold. BL represents the boundary of an object. An object is surrounded by BL.

In this article, static sensors are divided into different types according to their sensing attributes. A sensor node whose sensing value is higher than the threshold is called Inside Node (IN). A sensor node whose sensing value is lower than the threshold is called Outside Node (ON). BL separates the object and its surrounding.

As shown in Figure 2, all blue and black nodes that are located in the yellow object region have sensed the object, and they are INs. On the contrary, all the green nodes that are located on the outside of the object region are ONs.

Definition 2 (inside boundary node (IBN)). A node belongs to IN and there are both inside nodes and outside nodes on its one-hop neighbors.

Definition 3 (outside boundary node (OBN)). A node belongs to ON and there are both inside nodes and outside nodes on its one-hop neighbors.

All of IBNs belong to IBs, and all of OBNs belong to OBs. IBN and OBN are considered to be more close to the BL, because, around them, there exist different types of static sensors.

2.2. Spatial Interpolation. Spatial interpolation algorithms are widely used in geographic information systems like disaster risk analysis, agriculture, and event prediction particularly. To predict the condition of an unknown object, a spatial interpolation algorithm, namely, *Inverse Distance Weighted* (IDW), is applied in this article.

Assume that p represents a position which we want to acquire its estimated value $f(p)$, and $f(p)$ is influenced by all the points that exist a true sensing value. D_i is the distance between p and p_i and μ is a parameter that influences the

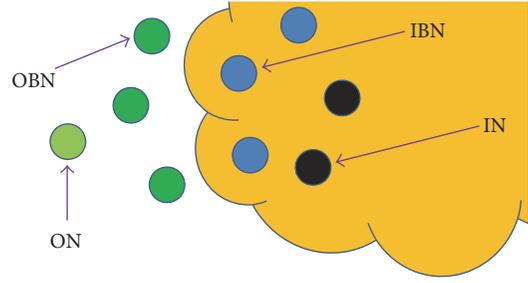


FIGURE 2: An example of static sensors distinction near the object boundary.

result of interpolation; it is usually set to 2. The influence of p_i to p is inversely proportional to D_i :

$$D_i = \sqrt{(x_p - x_{p_i})^2 + (y_p - y_{p_i})^2}, \quad (2)$$

$$f(p) = \frac{\sum_{i=1}^n (z_i / (D_i)^\mu)}{\sum_{i=1}^n (1 / (D_i)^\mu)}. \quad (3)$$

The main steps of *Inverse Distance Weighted* are listed:

- (i) Calculate the distance D_i between the point p and all known points.
- (ii) Calculate each point's weight $1/(D_i)^\mu$, which is the reciprocal of distance between p and p_i .
- (iii) Calculate the predicted value of the point p according to formula (3).

2.3. Planar Graph. The planar graph is a kind of graph which consists of three parts: vertexes, edges, and faces. In the planar graph, each face is a closed polygon that consists of N vertexes ($N \geq 3$). In WSNs, the planar graph is usually applied to organize the sensor network, and there have developed many kinds of planarization algorithms. The *Gabriel Graph* (GG) and the *Relative Neighborhood Graph* (RNG) are two well-known kinds of the planar graph as presented in [18] and [19], respectively.

Figure 3 shows the rule of RNG and GG. In RNG, if a node w is located in the intersection region of circles about u and v with radius $d(u, v)$, then the edge $e(u, v)$ is deleted and it builds the edge $e(u, w)$, $e(v, w)$ separately. In GG, if a node w is located in the circle about the midpoint of u and v with diameter of $d(u, v)$, then the edge $e(u, v)$ is deleted and separately builds the edge $e(u, w)$, $e(v, w)$.

Based on the planar graph, we define the following.

Definition 4 (boundary face (BF)). A polygon area is divided by the planar graph which contains both IBN and OBN on its vertexes.

Definition 5 (boundary area (BA)). It is a continuous area that consists of all BFs.

An example is shown in Figure 4. The yellow area is the object. All blue and black nodes represent IN. All green nodes

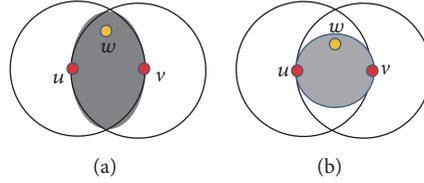


FIGURE 3: (a) shows the mechanism of generating the edge in RNG. If there exists a node w in the gray area, the edge between u and v will be broken and they connect with w , respectively. (b) shows the mechanism of generating the edge in GG. If there exists a node w in the gray area, the edge between u and v will be broken and they connect with w , respectively.

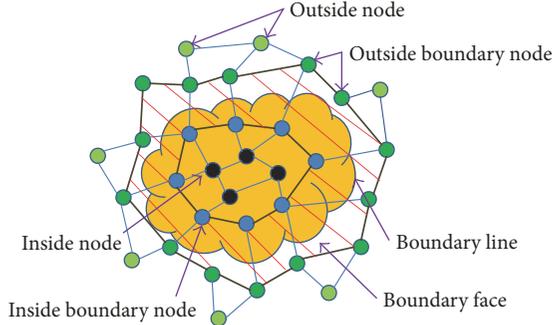


FIGURE 4: An example of object detection model.

represent ON. All polygon regions with both IBN marked by blue color and OBN marked by deep green color are named BF and they form the BA which is marked by red lines. BL is the curve that surrounds the yellow region.

3. Sensory Data Interpolation

In this section, the strategy for initializing the networks is presented. Planar graph algorithms are always applied to build topology networks for data transmission and network dividing. Spatial interpolation algorithms are applied to predict the object boundary.

3.1. Static Sensor Data Gathering. Static sensors are deployed in the network region at first. Once deployed, their positions are fixed. WSNs are often looked on as a self-organization network [20]. Static sensors communicate with their nearby nodes; then those sensors can transmit their position information to the base station leveraging network topology [21].

Static sensors are limited by energy storage. To reduce energy consumption, sensing time slot is introduced. At first time slot, static sensors start to sense their surroundings and transmit data to the base station. After that they stop sensing until it comes to the next sensing time slot.

Different from dense networks with so many sensors deployed in network region, this article works on a sparse network which has deployed less static sensors relatively. So there are some sensing holes in the region. In the proposed strategy, static sensors are on a small scale deployment and we gather all sensors' data at each time slot, instead of partly gathering sensory data as mentioned in many researches.

We do not focus on static sensors' data transmission strategy mainly because many researches have worked on

data transmission routing problem. In this section, we apply planar graph algorithms for building a routing map for static sensors and divided the network region. Different from the grid dividing mechanism in [22, 23], the network in this article is divided into many small faces. In each face region, edge connects two nodes within a one-hop communication range. Each sensor in this map could transmit its position information data and sensory data by multihop network to base station.

3.2. IDW Spatial Interpolation Mechanism. Once all sensors transmit their sensory data to the base station, then the data will be analyzed in following steps:

- (i) Based on planar graphs, a coarse boundary region BA is generated by BFs. The whole network region is divided into many subregions and static sensors are the vertexes of subregions according to planar graphs. Then BFs which have both IBNs and OBNs on their vertexes are selected and BA consists of all those BFs.
- (ii) Based on the real data sensed by all static sensors, we apply *Inverse Distance Weighted* to generate a predicted value for each point that has no real sensing data. According to the interpolation result, we generate a curve called predicted BL where all the interpolation values are equal to the threshold.

4. Routing for Mobile Sensors

It is thought that sensing holes exist in the network because of the sparse static sensors deployment. After a predicted BL is generated, to reflect the truth in the curve and make a more precise object boundary area, mobile sensors are applied to round the curve. In this section, we propose a strategy of stops selection at first. The mobile sensors deployment and routing strategy are presented then. Finally, the object boundary area precision is discussed.

4.1. Stop Stations Selection. Based on a predicted BL, we try to get the real data on it. Traversing and sensing the whole predicted BL are unpractical, and it is a common sense that trying to traverse some representative positions is effective. Mobile sensors are also with the ability of sensing and a limited sensing range. In the proposed strategy, assume that sensing range for mobile sensors is R . To find some suitable stops that could mainly reflect the true situation of the curve, the distance between two stops should be appropriate. We

Require:

- (i) D_i : Distance between a stop station s_i and first stop s_1 .
- (ii) R : sensing range of mobile sensors.
- (iii) L : The predicted BL where a series of stops would be selected

Ensure:

- (i) A series of stop stations on curve L .
- (1) Randomly choose the first stop station s_1 .
- (2) Randomly choose the second stop station s_2 from a point where points are in a $0.9R$ to R range from s_1 .
- (3) initial $D_i = 2R$
- (4) initial $i = 3$
- (5) **while** $D_{i-1} > R$ **do**
- (6) find a point set PS_i where points are in a $0.9R$ to R range from s_{i-1} .
- (7) **for** $j = 2$ to $(i - 1)$ **do**
- (8) $sPS_i =$ a subset of PS_i where $D_{ij} > R$
- (9) **end for**
- (10) Randomly choose a stop station s_i from selected sPS_i
- (11) **end while**
- (12) remove the stops which are so close to the already existed stops or static sensors from the stops set.

ALGORITHM 1: Stop station selection.

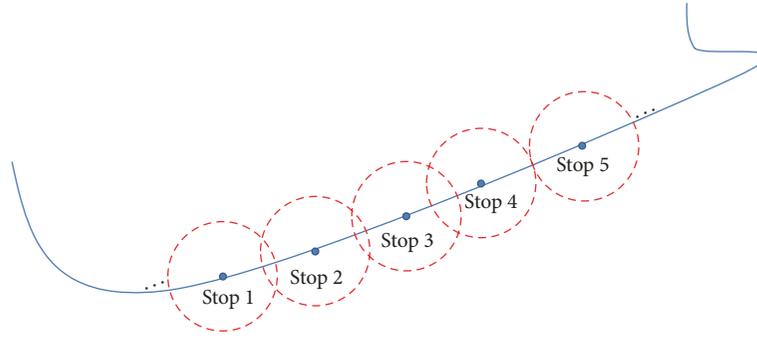


FIGURE 5: An example of stop selection. Each stop is selected and there is a suitable distance range between two adjacent stops.

suggest the distance between two adjacent stop stations is within the scope of $0.9R$ and R . As shown in Figure 5, it guarantees that the curve is covered completely. And a suitable distance can also reduce overlap.

Algorithm 1 is proposed to select stop stations. At first, an initial stop point s_1 is randomly selected in L , and the second point s_2 which is in $0.9R$ to R range far from s_1 is chosen (lines 1-2). Then a stop position s_i is picked from the point set PS_i generated according to s_{i-1} (lines 6-10). PS_i is a point set in which points are in $0.9R$ to R range far from s_{i-1} . After selection from PS_i , some points remain which are not so close to already existing stop points ($D_{i,j} > R$) (lines 7-9). Then it randomly chooses a point remaining in PS_i (line 10). When a set of stop points have been selected, then it checks whether a stop is within the sensing range of another point or sensors where sensory data exists already (line 12).

4.2. Mobile Sensor Determination and Routing Schemes. In this part, we discuss how to deploy mobile sensors and optimize the routing for them. In some applications, not only energy efficiency, but also the time constraints are taken into consideration.

4.2.1. Mobile Sensor Determination. As a set of stop points have been selected, then we deploy mobile sensors to traverse all those points in order to get the real sensory data. Time constraints are taken into consideration firstly. We assume that time spending on sensing of a sensor's surroundings is instant compared with moving. So we only concern about the time consumption on mobile sensor's movement. Based on this assumption, we propose a simple method by deploying a certain number of mobile sensors to ensure that all of the stop stations could be traversed on time.

At first, an estimated distance D_t which links all of the stop stations is generated. D_t is generated by choosing a stop station and linking it with its nearest stop, then the next stop also links with its nearest stop until all stops have been linked together. Then based on the speed of mobile sensors V_m , an estimated total time $T_t = D_t/V_m$ is calculated. According to the time constraint T_c , we get a minimum number of mobile sensors $N_s = \lceil T_t/T_c \rceil$.

4.2.2. Mobile Sensor Routing Schemes. In terms of the number of mobile sensors, in this part, a mechanism based on a heuristic algorithm is proposed. The target of this proposed scheme is to find optimal routings for all mobile sensors.

Require:(i) BA_C : Current object boundary area.**Ensure:**(i) BA_N : A new object boundary area.

- (1) According to the data gathered by both static sensors and mobile sensors, interpolate to the BA_C .
- (2) Applying Algorithm 1 to find a group of new stop stations.
- (3) Deploying a suitable number of mobile sensors.
- (4) Routing for each mobile sensor.
- (5) mobile sensors move to stop stations and get the sensory data there.
- (6) generating the BA_N based on planar algorithms.

ALGORITHM 2: Boundary area precision.

It requires that energy consumption and time consumption are similar for each mobile sensor. It can be formulated as a multiobjective optimization problem. Some parameters are represented as follows:

- (i) D_i is the moving distance of a mobile sensor M_i .
- (ii) N_{stop} is the number of stop stations that a mobile sensor M_i would traverse.
- (iii) $EM_i(D_i, v)$ is the energy consumption in movement in distance D_i with speed v . We assume that all those sensors have same movement ability.
- (iv) $ES_i(\text{data}_S, N_{\text{stop}})$ is the energy consumption in data transmission. data_S is the package of sensory data generated in a stop.
- (v) T_i is time a mobile sensor would spend. $T_i = D_i/v$. The constraint is $T_i < T_c$.
- (vi) $V(EM_i)$ represents the variance of energy consumption in a sensor's movement.
- (vii) $V(ES_i)$ represents the variance of energy consumption in a sensor's sensing.

Output is a tuple which includes a series of sequence denoted as $\{\text{sequence}_1, \text{sequence}_2, \dots, \text{sequence}_i\}$ where sequence_i represents an ordered stop points array that a mobile sensor M_i would traverse.

The fitness function is defined as follows:

$$\text{fitness} = k1 * V(EM_i) + k2 * V(ES_i), \quad (4)$$

where $k1$ and $k2$ are weights of two subobjects. They can adjust the importance of two subobjects. By adjusting the value of $k1$ and $k2$, we can determine the importance of two subobjects.

To solve this multiobjective and multiconstrained optimization problem, heuristic algorithm like ACO can be applied to solve this kind of issues [24, 25]. ACO is an evolutionary algorithm inspired by ants for finding foods behavior. It simulates the process of biological evolution to find the optimal solution. We apply it to find an optimal stops array for each mobile sensor. Through adopting ACO technique, an approximately optimal $\{\text{sequence}_1, \text{sequence}_2, \dots, \text{sequence}_i\}$ is found.

TABLE 1: Parameter settings.

Parameter name	Value
Region size	500 m × 500 m
Number of static sensors (n)	100–500
Radius of object (r)	40 m–120 m
Velocity of mobile sensors (V)	5 m/s
Energy consumption in data gathering (ECS)	1 unit/stop
Energy consumption in moving (ECM)	20 units/m
Sensing range of sensors (R)	10 m
Range of distance between stops (d)	0.5 R –1.5 R

4.3. Object Boundary Area Precision. Mobile sensors traverse along their own routings, and they gather the sensory data of all stops located in the predicted BL. The sensory data reflects the truth in stops and it may be different with their predicted values in some stops. With more and more stops being found, We use planar algorithms to redivide the networks again; then we could get a new BA. The faces which are closed to the BL could be smaller and thus the new BA could be smaller and the shape of new BA could be more precise to reflect the object boundary.

In order to reflect the true condition around the object boundary, the previous steps are reused to get a more and more precise BL. The details are represented in Algorithm 2. Firstly, it uses existing sensory data to interpolate in current BA (line 1). Then it selects a set of stop stations in the new BL according to Algorithm 1 and deploys some mobile sensors to traverse those stop stations (lines 2–5). Based on new sensory data gathered by mobile sensors, the planar algorithm is applied to divide the network region and a more precise BA is generated (line 6).

5. Implementation and Evaluation

The prototype has been implemented in a Java program and a hybrid WSN is constructed. Experimental settings and results are presented as follows.

5.1. Experimental Settings. The parameter settings for our experiments are presented in Table 1. There is a

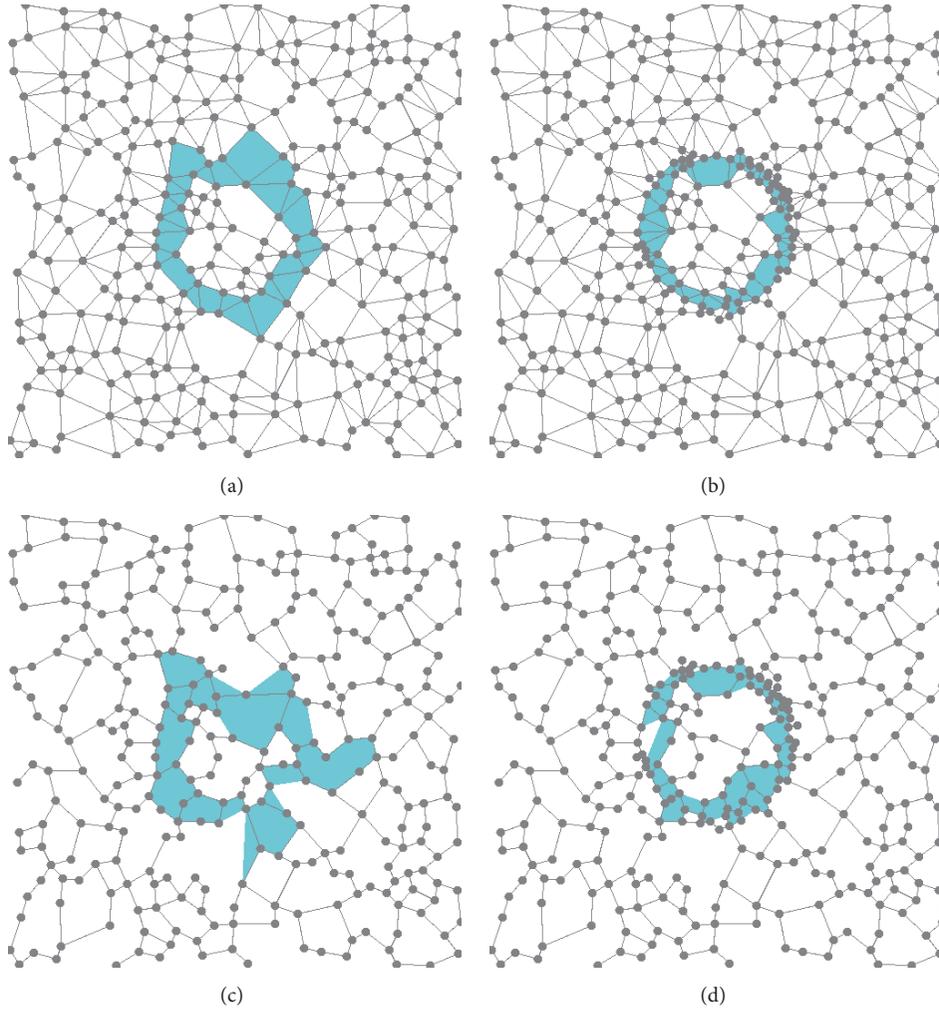


FIGURE 6: Object boundary regions of proposed mechanism. (a) shows the initial BA based on GG. (b) shows the BA after 10 time slots based on GG. The blue region in (c) shows the initial BA based on RNG and (d) shows the RNG-based BA after a period of time.

500 m \times 500 m network region and the number of static sensors ranges from 100 to 500 in increments of 100. We assume that the object is a circle. And there are three different types of objects whose radius range from 40 m to 120 m increased by 40 m. The sensing value in the center of the object is 300 and the threshold of the object is set to 216.5. That means a position whose sensing value is higher than the threshold inside the object. In this section, the performance of the proposed mechanism is verified by different restricted conditions.

5.2. Boundary Area Precision Analysis. In this part, we discuss the precision of BA. The impact of different planar algorithms, density of static sensors, object sizes, and distance range between stops is taken into consideration in this paper. We analyze the impact of those four factors in this part.

5.2.1. Impact of Planar Algorithm. Two different planar algorithms GG and RNG are applied in this article. Figure 6 shows the variation of the shape of BA. There are 300 static sensors deployed in the area, and the radius of the object is 80 m. The

blue area marks the BA of object. Compared with the initial network, the BAs with iterations are more regular and could more accurately reflect the real boundary region. And it is clear that GG and RNG result in different shape of BA. In fact, GG and RNG organize the network topology in different way, and the size and shape of their faces are different from each other. This results in a difference between the two algorithms.

5.2.2. Impact of Different Amount of Static Sensors. In this part, both GG and RNG are applied to initialize the network. And the number of static sensors n ranges from 100 to 300. Figures 7 and 8 show the variation of the size of BA and the number of stop stations. In Figure 7, with the increase of time slots, the size of BA is decreased in all of the curves. And after a period of time, the tendency of each curve holds steady. And with the increment of n , the size of BA is decreasing within GG and RNG, respectively.

Figure 8 shows the increment of stop stations. The number of stations increases but the growth rate decreases. The reason is that more and more selected stops are located in the predicted BL and they have covered the predicted BL

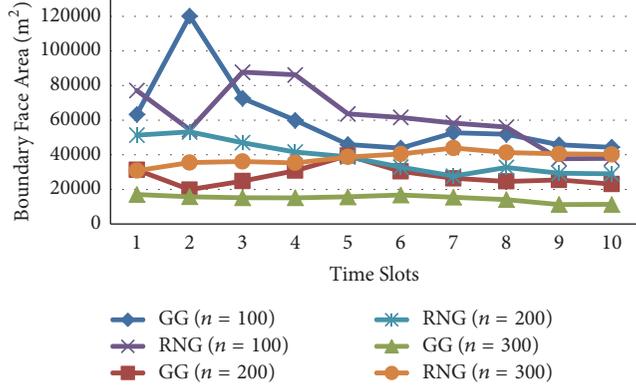


FIGURE 7: Comparison of the size of BA when different initializations and different n are applied.

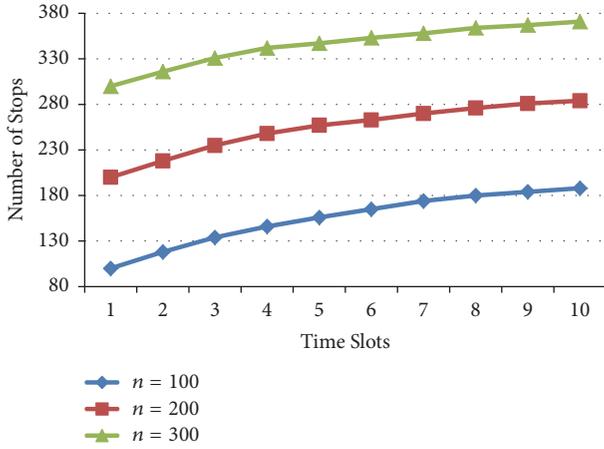


FIGURE 8: Comparison of the number of stop stations when different n are applied.

almost. In fact, when the number of stops is increasing, the size of boundary faces near the BL is increasing, and the number of boundary faces is decreasing. Finally, BL is covered by a narrow boundary region.

5.2.3. Impact of Object Size. Figures 9 and 10 show the simulation results. The radius of objects is set to 40 m, 80 m, and 120 m, respectively. Figure 9 shows the size of BA with the increment of time slots. Making a comparison under the same object size, GG and RNG perform differently in the BA's acreage. It is obvious that the size of BA when GG applied is mostly smaller than that of RNG in each time slot.

Figure 10 is the result of the increment of stop stations. The number of stop stations increases with the increasing of time slot. The growth rate is decreasing but growth rates are similar in all those curves. The scales of stop stations' number are different when the radius of objects are different. The smaller the object radius is, the less stop stations it selects in the predicted BL.

5.2.4. Impact of Distance between Stops. Figures 11 and 12 show the simulation results. The distance between stops ranges from about $0.5R$ to $1.5R$ in increments of $0.5R$. In

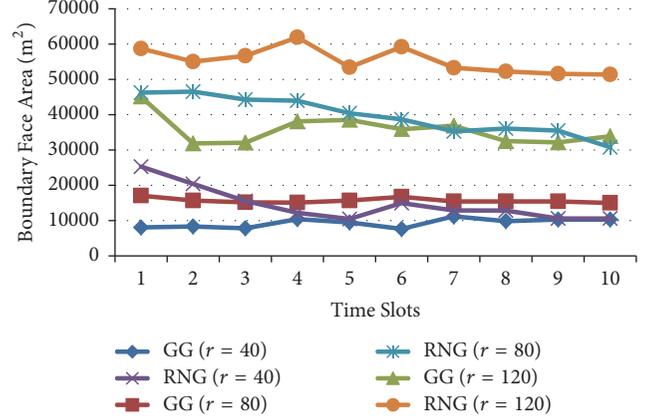


FIGURE 9: Comparison of the size of BA when different initializations and different r are applied.

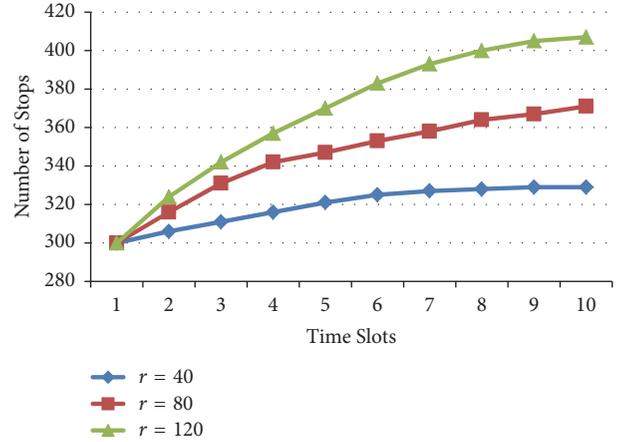


FIGURE 10: Comparison of the number of stop stations when different r are applied.

Figure 11, with the increment of time slot, the size of BA is decreasing in all of the curves. And after a period of time, the tendency of each curve holds steady. In GG, the range of BA's acreage is increasing when the distance between stops increases. In RNG, when the distance between two nearest stop stations is on a range of R , the range of BA's acreage is the smallest. It is obvious that different distance ranges between stops have an influence on the size of BA.

Figure 12 shows the number of stop stations. In each curves, the number of stops increases with the time going on. And the growth rate of stop stations number is decreasing. When the distance is on a range of $0.5R$, the increasing of stops number are less than that of R and $1.5R$. And the tendency of the increment of stops when the distance between two nearest stops is R or $1.5R$ are similar. In fact, if there exists many selected stops near BL, it is hard to select more stops because most of related region have been detected and there is no need to detect them again.

5.3. Boundary Precision Analysis. In this part, the precision of BL is shown and we compare the proposed mechanism with COBOM [12]. The object boundary precision is used

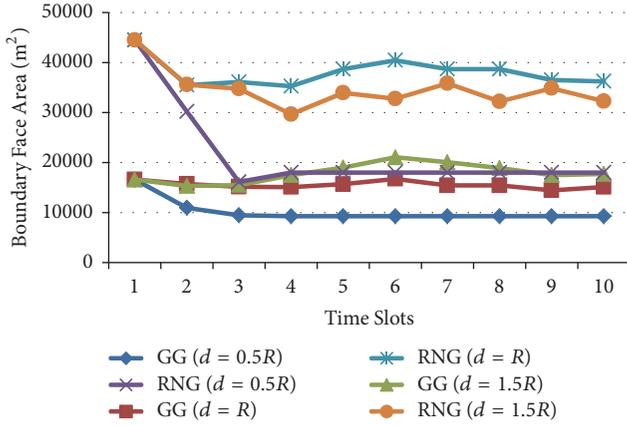


FIGURE 11: Comparison of the size of BA when different initializations and different d are applied.

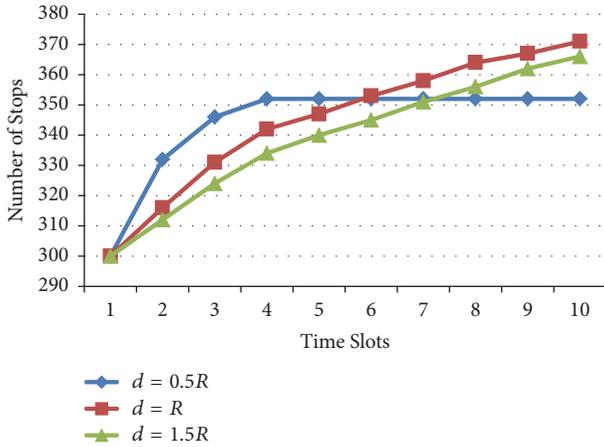


FIGURE 12: Comparison of the number of stop stations when different d are applied.

to make comparison. The object boundary precision in this article is the proportion of boundary nodes which are close to the object boundary. If more boundary nodes are close to the real boundary, the boundary precision is higher.

Figure 13 shows the result of boundary precision under different density of node deployment. The proposed mechanism has a higher boundary precision than that of COBOM because the mechanism in this article can get the sensory data from sensing holes. Mobile sensors are deployed to move to the area without static sensors. COBOM only allows a small set of sensors to report their data; however, the network is sparse and a representative node may be far from the real boundary. The curves of COBOM have no changes. It is because the object boundary is stable. In each time slot, the status of boundary nodes is not changed. So the boundary precision is not changed once the boundary nodes are selected in COBOM.

Figure 14 shows the result of boundary precision with different distance between stops. We randomly deployed 300 sensor nodes in a $500\text{ m} \times 500\text{ m}$ region. The distance d between two nearest stops ranges from $0.5R$ to $1.5R$. With the

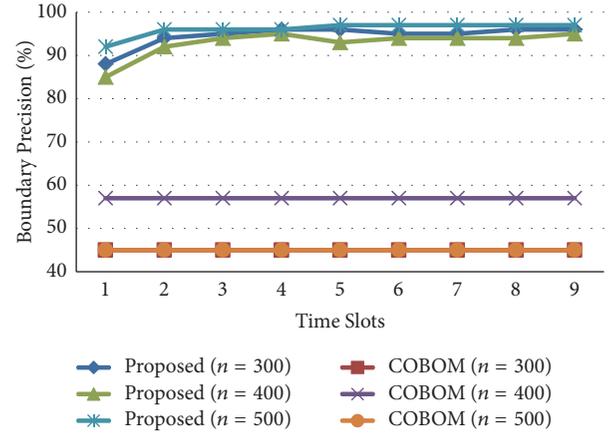


FIGURE 13: Comparison of the boundary precision for proposed mechanism and COBOM when n is set to 300, 400, and 500.

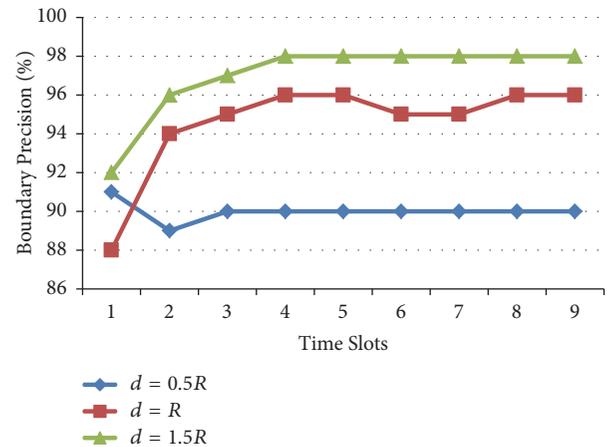


FIGURE 14: Comparison of the boundary precision when different d are applied.

increment of time durations, the boundary precision keeps stable in all curves. And they are on a high level from 90% to 98%. In fact, a well coverage with stops could increase the accuracy of detection.

5.4. Mobile Sensor Performance Analysis. In this section, we analyze the performance of mobile sensors. After network initialization, we select some stop stations on the predicted BL. Assume that time constraint is T , that means we should collect all of the sensory data before T . We estimate a length L for traversing all those stop stations, and a predicted minimum number of mobile sensors is calculated based on the velocity of mobile sensors. Then this problem can be looked on as a multiple Traveling Salesman Problem, and ACO is applied to solve this problem; the objective function is to minimize the total length for all mobile sensors and to balance the length for each mobile sensor, thus balancing the time consumption.

The time constraint T is set to 50 s, and we have known that $V = 5\text{ m/s}$. According to T and V , we implement three different sizes of object about 40 m, 80 m, and 120 m. The

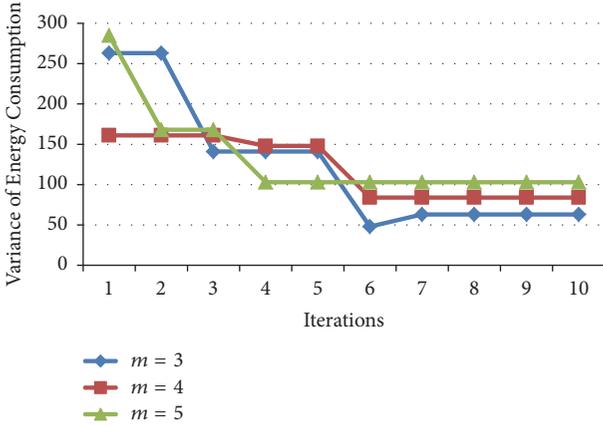


FIGURE 15: Comparison of the variance of energy consumption when m is set to 3, 4, and 5.

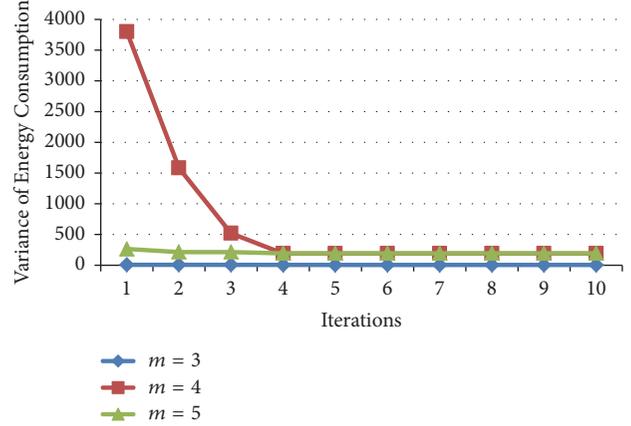


FIGURE 17: Comparison of the variance of energy consumption when m is set to 3, 4, and 5.

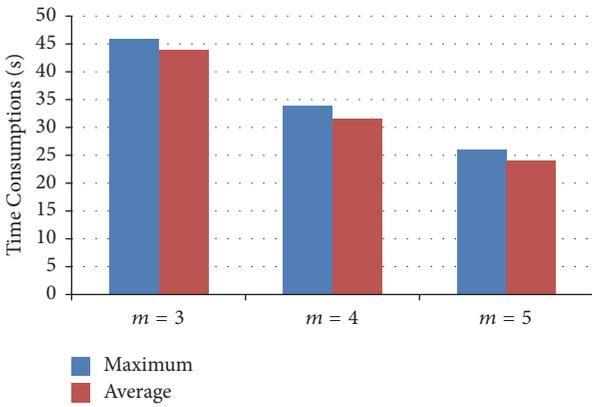


FIGURE 16: Comparison of the time when m is set to 3, 4, and 5.

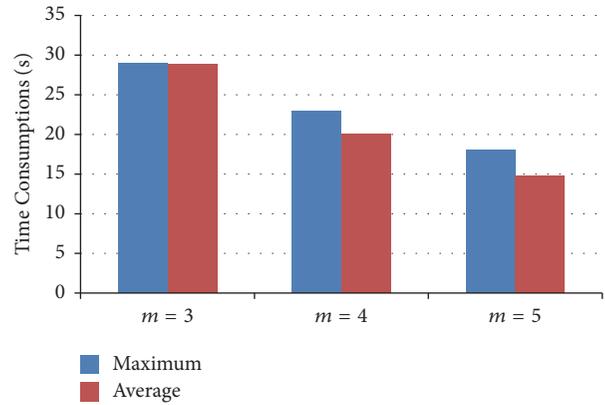


FIGURE 18: Comparison of the time when m is set to 3, 4, and 5.

number of mobile sensor m ranges from 1 to 5. Figures 15–20 show the results of implementations.

In Figures 15, 17, and 19, the changes of variance of energy consumption are shown. It is clear that the variance decreases and trends to a stable value. That means an optimistic scheduling for each mobile sensor is generated.

In Figures 16, 18, and 20, the maximum and average time consumption for mobile sensors are shown with different objects ranging from 40 m to 120 m. It is obvious that, according to the proposed scheme, there must be a suitable number of mobile sensors that could satisfy the time constraint T . With the increasing number of mobile sensors deployed in the network, the maximum and average time consumption would decrease.

6. Related Works

There has been long-term progress on object detection and tracking research in WSNs. Recently, fog computing framework has been more and more popular in IoT applications [26–29]. In [30], a mobile WSN is applied to intrusion detection, and fog computing is applied to improve the network performance. A three-tier model is proposed where

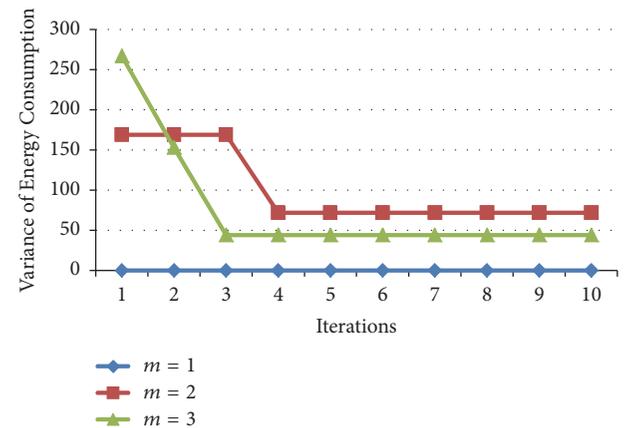


FIGURE 19: Comparison of the variance of energy consumption when m is set to 1, 2, and 3.

WSN is a bottom tier. Fog servers are located in the middle tier to manage the WSN and be connected to the cloud. Fog computing builds a flexible framework for gathering data. It is more close to the data source so that latency could be reduced.

In [31], a toxic gas boundary area detection scheme is proposed whose target is to detect the boundary area

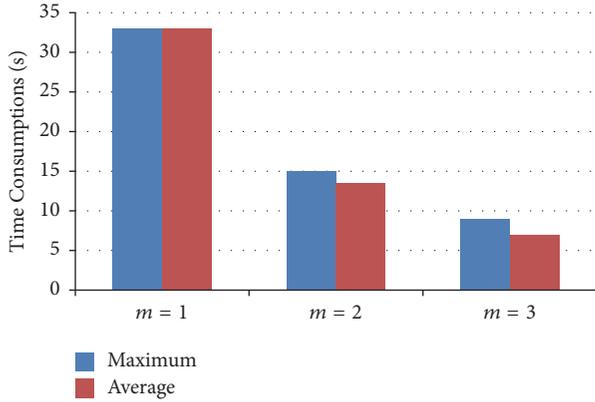


FIGURE 20: Comparison of the time when m is set to 1, 2, and 3.

rather than a single boundary line. Planarization algorithm is applied into the network region and it is divided into many small faces. Then based on inner-boundary node (IBN) and outer-boundary node (OBN), they identify a boundary area. IBN is a node where the reading of the toxic gas is higher than the alarm threshold. OBN is a node where the reading of the toxic gas is lower than the alarm threshold. The characteristic of boundary area is that it has both IBNs and OBNs on its vertices. But it does not move forward to detect the holes inside the faces. In this article, we concentrate on detecting the sensing holes by mobile sensors according to a predicted boundary line and the shape of boundary area is more precise.

In [10], an energy-efficient algorithm for detection and monitoring object boundary called COBOM is proposed. They only choose a part of nodes to report to the sink node. If the current reading of a sensor is different from its previous reading, it broadcasts its reading and ID information. A node that receives the reading and information will store them in a BN-array. If its BN-array exists in different reading, it becomes a boundary node (BN). Only a part of BNs are selected to report message to the sink node, which are called representative nodes. Those representative nodes also suppress their neighbor nodes so that no significant information is discarded. In DEMOCO [32], if a node receives a message containing the same status, it ignores the message. If it receives a different message that is also different from at least one of its neighbor nodes, it is called a boundary node. Different back-off time is assigned to different boundary nodes, and if a boundary node is back early in short duration, it will suppress other nearby boundary nodes in sending messages. They are called representative nodes (RNs). Compared with COBOM, DEMOCO produces fewer BN and RN nodes and it performs better than COBOM.

Sleep scheduling mechanism is proposed to be an energy-efficient way in WSNs. In [11] an energy-efficiency scheme for monitoring a large-scale object is used. At first sensor nodes send message to their neighboring nodes. A node that receives a message including different status and has the shortest distance with the sender node is called boundary node (BN). Then master boundary nodes (MBNs) are selected from BNs based on the received event messages.

MBNs are the same concept as RN in COMBO and DEMOCO. Based on the above steps, it predicts next boundary by collaborating between MBNs. A boundary line is recognized by linked MBNs which is named the current boundary line. According to object diffusing, they draw a predicted next boundary. Sleeping sensor nodes in the next boundary zone waked up to send report. In this scheme, only the sensor nodes near the boundary area are activated to save energy. And next boundary is predicted to a precise object boundary detection.

The authors in [33] presented a sleep scheduling scheme for toxic gas monitoring. A sensor node inside the gas leakage area is selected as the zone head based on the largest residual energy by restricting flooding of the head selection message within the zone. CKN algorithm is applied to ensure global connectivity. Each awake node reports the sensing status and the grid centers covered by itself until the zone head. A awake node sends a wake-up message which contains its position, node ID, and the coordinates of the grid-points covered by itself to its one-hop sleep-neighbor. The sleep node is estimated and wakes up according to its neighbor nodes status.

In [9] a mobile element path planning for time-constrained data gathering scheme is proposed. Data produced by measuring needs to be delivered to predefined sink within a given time interval from the beginning of the measurement. Mobile elements travel along a predefined path and collect data. Every node should be visited, and a node-disjoint scheduling scheme is presented for searching an approximately optimal path. In our previous work [34], a WSN region is divided into many small areas: each area should be visited by mobile sensors for gathering data. Multiple mobile sensors are deployed in network: each of them is responsible for a certain subregion, and heuristic algorithms are applied for routing.

In [35] ant colony optimization (ACO) is introduced to schedule mobile sensors in consumer home automation networks. The network is partitioned into several clusters and only one sensor node serves as a CH in each cluster. A node that has a relative higher residual energy is called CH. It forms a center of a cluster by broadcasting message to other nearby sensors and gathering their data. Authors in this work only applied one mobile sensor to traverse those CHs and collect data. The mobile sensor relieves excessive consumption of part nodes and prolongs the lifetime of the consumer home network. ACO is utilized to plan the mobility path for the mobile sensor. In our work, we apply more than one mobile sensor considering the timeliness.

7. Conclusion

In this article, we propose to detect object boundary region through applying mobile sensors. The network is divided by planar algorithms at first. An estimated object boundary is derived through applying the interpolation algorithm. To examine whether the boundary reflects that fact, candidate sensing locations are discovered and traversed by mobile nodes for gathering sensing data. The heuristic algorithm (i.e., ACO) is applied to generate optimal paths for mobile

sensors. Experimental results show that the proposed mechanism can get a precise object boundary region and can balance the energy consumption and time consumption for mobile sensors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported partially by the National Natural Science Foundation of China (Grant no. 61772479) and by the Fundamental Research Funds for the Central Universities (China University of Geosciences (Beijing), China).

References

- [1] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "FogRoute: DTN-based data dissemination model in fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 225–235, 2016.
- [2] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, 2016.
- [3] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.
- [4] D. V. Manatakis and E. S. Manolakis, "Estimating the Spatiotemporal Evolution Characteristics of Diffusive Hazards Using Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 9, pp. 2444–2458, 2015.
- [5] T. Qiu, R. Qiao, and D. Wu, "EABS: an event-aware backpressure scheduling scheme for emergency internet of things," *IEEE Transactions on Mobile Computing*, 2017.
- [6] W.-R. Chang, H.-T. Lin, and Z.-Z. Cheng, "CODA: a continuous object detection and tracking algorithm for wireless ad hoc sensor networks," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 168–174, January 2008.
- [7] S. Park, H. Park, E. Lee, and S.-H. Kim, "Reliable and flexible detection of large-scale phenomena on wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 6, pp. 933–936, 2012.
- [8] S. Imran and Y.-B. Ko, "A continuous object boundary detection and tracking scheme for failure-prone sensor networks," *Sensors*, vol. 17, no. 2, article 361, 2017.
- [9] K. Almi'ani, A. Viglas, and L. Libman, "Mobile element path planning for time-constrained data gathering in wireless sensor networks," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, AINA2010*, pp. 843–850, Australia, April 2010.
- [10] C. Zhong and M. Worboys, "Energy-efficient continuous boundary monitoring in sensor networks," *Tech. Rep.*, 2007.
- [11] S. Park, S.-W. Hong, E. Lee, S.-H. Kim, and N. Crespi, "Large-scale mobile phenomena monitoring with energy-efficiency in wireless sensor networks," *Computer Networks*, vol. 81, pp. 116–135, 2015.
- [12] X. Ji, H. Zha, J. J. Metzner, and G. Kesidis, "Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks," in *Proceedings of the 2004 IEEE International Conference on Communications*, pp. 3807–3811, fra, June 2004.
- [13] H.-J. Lee, M. T. Soe, S. H. Chauhdary, S. Rhee, and M.-S. Park, "A data aggregation scheme for boundary detection and tracking of continuous objects in WSN," *Intelligent Automation and Soft Computing*, vol. 23, no. 1, pp. 135–147, 2017.
- [14] Y. Zhang, X. Zhang, W. Fu, Z. Wang, and H. Liu, "HDRE: coverage hole detection with residual energy in wireless sensor networks," *Journal of Communications and Networks*, vol. 16, no. 5, pp. 493–501, 2014.
- [15] F. Yan, A. Vergne, P. Martins, and L. Decreusefond, "Homology-based distributed coverage hole detection in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1705–1718, 2015.
- [16] S. Das, I. Banerjee, and T. Samanta, "Sensor localization and obstacle boundary detection algorithm in WSN," in *Proceedings of the 3rd International Conference on Advances in Computing and Communications*, pp. 412–415, August 2013.
- [17] K. M. Alam, J. Kamruzzaman, G. Karmakar, and M. Murshed, "Dynamic adjustment of sensing range for event coverage in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 46, pp. 139–153, 2014.
- [18] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, no. 3, pp. 259–278, 1969.
- [19] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.
- [20] T. Qiu, A. Zhao, F. Xia, W. Si, and D. O. Wu, "ROSE: Robustness Strategy for Scale-Free Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2944–2959, 2017.
- [21] T. Qiu, R. Qiao, M. Han, A. K. Sangaiah, and I. Lee, "A Lifetime-Enhanced Data Collecting Scheme for the Internet of Things," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 132–137, 2017.
- [22] Z. Zhou, C. Du, L. Shu, G. Hancke, J. Niu, and H. Ning, "An Energy-Balanced Heuristic for Mobile Sink Scheduling in Hybrid WSNs," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 28–40, 2016.
- [23] S. Sharma, D. Puthal, S. Tazeen, M. Prasad, and A. Y. Zomaya, "MSGR: A Mode-Switched Grid-Based Sustainable Routing Protocol for Wireless Sensor Networks," *IEEE Access*, vol. 5, pp. 19864–19875, 2017.
- [24] Y. Lin, J. Zhang, H. S.-H. Chung, W. H. Ip, Y. Li, and Y.-H. Shi, "An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 3, pp. 408–420, 2012.
- [25] W. Liu, S. Li, F. Zhao, and A. Zhen, "An ant colony optimization algorithm for the multiple traveling salesmen problem," in *Proceedings of the 2009 4th IEEE Conference on Industrial Electronics and Applications, ICIEA 2009*, pp. 1533–1537, China, May 2009.
- [26] A. M. Rahmani, T. N. Gia, B. Negash et al., "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.
- [27] J. Liu, J. Li, L. Zhang et al., "Secure intelligent traffic light control using fog computing," *Future Generation Computer Systems*, vol. 78, pp. 817–824, 2018.
- [28] B. R. Stojkoska, K. Trivodaliev, and D. Davcev, "Internet of things framework for home care systems," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 8323646, 2017.

- [29] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On Reducing iot Service Delay via Fog Offloading," *IEEE Internet of Things Journal*, 1 page.
- [30] Q. Yaseen, F. Albalas, Y. Jararweh, and M. Al-Ayyoub, "A fog computing based system for selective forwarding detection in mobile wireless sensor networks," in *Proceedings of the 1st International Workshops on Foundations and Applications of Self-Systems, FAS-W 2016*, pp. 256–262, Germany, September 2016.
- [31] L. Shu, M. Mukherjee, and X. Wu, "Toxic gas boundary area detection in large-scale petrochemical plants with industrial wireless sensor networks," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 22–28, 2016.
- [32] J.-H. Kim, K.-B. Kim, S. H. Chauhdary, W. Yang, and M.-S. Park, "DEMOCO: Energy-efficient detection and monitoring for continuous objects in wireless sensor networks," *IEICE Transactions on Communications*, vol. E91-B, no. 11, pp. 3648–3656, 2008.
- [33] M. Mukherjee, L. Shu, L. Hu, G. P. Hancke, and C. Zhu, "Sleep Scheduling in Industrial Wireless Sensor Networks for Toxic Gas Monitoring," *IEEE Wireless Communications Magazine*, vol. 24, no. 4, pp. 106–112, 2017.
- [34] Y. Zhang, Z. Zhou, D. Zhao, M. Barhamgi, and T. Rahman, "Graph-based mechanism for scheduling mobile sensors in time-sensitive WSNs applications," *IEEE Access*, vol. 5, pp. 1559–1569, 2017.
- [35] J. Wang, J. Cao, B. Li, S. Lee, and R. S. Sherratt, "Bio-inspired ant colony optimization based clustering algorithm with mobile sinks for applications in consumer home automation networks," *IEEE Transactions on Consumer Electronics*, vol. 61, no. 4, pp. 438–444, 2015.



Hindawi

Submit your manuscripts at
www.hindawi.com

