

Research Article

Detecting Boundary Nodes and Coverage Holes in Wireless Sensor Networks

Li-Hui Zhao,^{1,2} Wenyi Liu,^{1,2} Haiwei Lei,^{1,2} Ruixia Zhang,^{1,2} and Qiulin Tan^{1,2}

¹Key Laboratory of Instrumentation Science & Dynamic Measurement, Ministry of Education, North University of China, Shanxi, Taiyuan 030051, China

²National Key Laboratory for Electronic Measurement Technology, North University of China, Taiyuan, Shanxi 030051, China

Correspondence should be addressed to Wenyi Liu; liuwynuc@126.com

Received 19 October 2015; Revised 18 January 2016; Accepted 18 January 2016

Academic Editor: Laurence T. Yang

Copyright © 2016 Li-Hui Zhao 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 emergence of coverage holes in wireless sensor networks (WSNs) means that some special events have broken out and the function of WSNs will be seriously influenced. Therefore, the issues of coverage holes have attracted considerable attention. In this paper, we focus on the identification of boundary nodes and coverage holes, which is crucially important to preventing the enlargement of coverage holes and ensuring the transmission of data. We define the problem of coverage holes and propose two novel algorithms to identify the coverage holes in WSNs. The first algorithm, Distributed Sector Cover Scanning (DSCS), can be used to identify the nodes on hole borders and the outer boundary of WSNs. The second scheme, Directional Walk (DW), can locate the coverage holes based on the boundary nodes identified with DSCS. We implement the algorithms in various scenarios and fully evaluate their performance. The simulation results show that the boundary nodes can be accurately detected by DSCS and the holes enclosed by the detected boundary nodes can be identified by DW. The comparisons confirm that the proposed algorithms outperform the existing ones.

1. Introduction

Wireless sensor networks are widely applied in many fields such as battlefield surveillance [1, 2], target tracking [2–4], environment monitoring [5], and health care [6, 7]. Most of the above applications are mission-critical and require complete coverage of the monitored area so that the events under observation can be precisely detected. However, these requirements constantly encounter the obstacles from coverage holes. On the one hand, coverage holes degrade the performance of WSNs, such as decreasing the connectivity, unbalancing the WSNs load, and aggravating the burden of hole border nodes in data forwarding. On the other hand, the appearance of coverage holes indicates the burst of special events, such as forest fire and earthquake [8]. Unfortunately, due to nonuniformity of deployment [9], failure of nodes [10], and obstacle of terrain [11], coverage holes are bound to appear and gradually increase in number and size. Thus, to alleviate the increase is stringently demanded [12]. However,

no matter which idea is adopted, optimizing routing protocols [13, 14] or reducing energy consumption [15], to identify coverage holes comes first and foremost.

Existing works take advantage of various characteristics of nodes in terms of geometrical morphology [4, 16–22], distribution [23, 24], and connectivity [8, 12, 25–30] to identify coverage holes. However, most of the algorithms require relatively high average node degrees [23, 24, 29], leading to a high communication overhead [25] in the process of searching the specific patterns [28, 29]. It is noted that although all the algorithms identify coverage holes by locating the nodes on the boundary of holes, none of them provides an approach for clustering the nodes except [25]. Regrettably the tremendous communication overhead in [25] badly shrinks the life cycle of WSNs.

In this paper, we tackle the problem of identifying coverage holes in various scenarios by detecting the nodes located on the border of holes. Our contributions are as follows. First, we formally define the notions of boundary nodes and

coverage holes. Second, we generalize our work into two questions: one is how boundary nodes are identified and the other is how coverage holes are detected. Correspondingly, we propose two algorithms, Distributed Sector Cover Scanning (DSCS) and Directional Walk (DW) algorithm. DSCS, motivated by the observation that a node can have its sensing area divided into different sectors by its neighbors, transforms the problem of identifying boundary nodes into judging whether all the sensing sectors of a node are covered by the adjacent neighbors of the node. DW, based on DSCS, is responsible for detecting coverage holes by connecting the boundary nodes that enclose the same coverage hole and clustering the nodes on the outer boundary of WSNs. Finally, we implement DSCS and DW and verify their correctness and robustness.

The remainder of this paper is organized as follows. Section 2 reviews related works. In Section 3, we present the net model and define coverage holes and boundary nodes. Section 4 presents the DSCS and DW algorithms in detail and states the related theoretical analysis to prove the correctness of the algorithms. Section 5 presents a part of the simulation results and compares the proposed algorithms with the existing ones. Finally, we conclude the paper.

2. Related Work

In this section, we recall the previous studies on the boundary node identification and coverage hole detection. The existing works can be classified into three categories: geometrical, statistical, and topological [25].

In general, geometry-based methods employ the geometrical tools to recognize the boundary nodes and coverage holes with the assumption that nodes are aware of their own locations. Fang et al. [16] proposed a two-step distributed approach, in which each node autonomously checks whether it is a stuck node serving the identification of coverage holes and then an alternative route is created for data transmission along the holes. Khedr et al. applied Barycentric technique to propose a decentralized algorithm in [17], in which each node can determine whether it is a boundary node by calculating the distance of the adjacent nodes. The scheme proposed by Deogun et al. identifies a boundary node by Choosing Good Neighbors (CGN) of a node and then checking whether the Good Neighbors can form a triangle enclosing the node in [18]. Simek et al. proposed Boundary Recognition using Cset (BRC) in [19] based on the improved CGN. The implementation of both CGN and BRC is restricted by the communication range of nodes. Ma et al. [20] utilized the characters of the circumcircle of a triangle to propose a simple distributed algorithm for boundary node identification. Wang and Ssu described a scheme in [21] that if all the intersection points of the sensing area of a node with those of other nodes can be covered by the sensing region of any other node, then the node is not a boundary node. Li and Zhang [22] used the properties of empty circle based on Delaunay triangulation [31] to discover boundary nodes. All the above-mentioned algorithms can recognize whether a node is located on the border of coverage holes. However,

few of them offered a clear scheme how coverage holes are identified.

Statistical methods usually identify boundary nodes by assuming that the nodes in WSNs abide by the probability distribution. Fekete et al. [23] proposed an algorithm to detect boundary nodes based on the fact that mostly the node degree of boundary nodes is less than that of the nonboundary nodes of WSNs. Although good results can be achieved when the node degree in WSNs is smooth, the method is prone to being invalid when the node degree fluctuates sharply. In their subsequent work, Fekete et al. [24] derived the location of boundary nodes from the restricted stress centrality of node. However, to achieve the desirable performance, both of the above methods demand such an unrealistic node density that the average node degree should be not less than 100.

Topology-based approaches, particularly applied to the location-free environment, recognize boundary nodes and coverage holes by exploiting the connectivity information of nodes. Funke et al. developed a distributed hole detection algorithm (DHD) that identifies boundary nodes by locating the broken contour in [8]. DHD firstly selects each node in the max independent set of WSNs as a seed to build its different layers of contours by connecting its neighbors that share the same hop distance, and then the algorithm identifies boundary nodes by locating the end points of a broken contour. On the basis of DHD, Chu and Ssu [26] proposed to recognize boundary nodes through creating the contour of each node based on the three-hop neighboring information of the node. Huang et al. [27] extended the idea of isocontour and proposed a scheme that uses the two-hop neighbors of any node to build the isocontour of it. The node is a boundary node if its isocontour is broken. Although the scheme does not require the location of node, it marks too many boundary nodes. Kröller et al. [28] identified coverage holes and the outer boundary of WSNs by searching a complex combinatorial structure, called flower, on the assumed prerequisite that the average node degree is more than 20. After that, Saukh et al. improved the flower pattern to recognize boundary nodes in [29], but both of them fail to detect some special patterns. Different from the above, Wang et al. [25] utilized the cut pair of the shortest path tree constructed by an arbitrary node in WSNs to identify boundary nodes, which can work well even with low average node degree. However, the high communication cost produced by repeatedly using the flooding way shrinks the lifetime of WSNs. To overcome this problem, Khan et al. proposed a hop-based approach in [12]. In this approach, each node determines whether it is a boundary node by checking whether the k -hop neighbors of a node can form an enclosed cycle of the node. Ghrist and Muhammad put forward a scheme [30] based on algebraic homology theory. However, the centralized design does not follow the trend of actual implementation in WSNs.

In this paper, we propose two distributed algorithms based on computational geometry, which can effectively prevent the drawbacks of the aforementioned approaches. In the first algorithm, each node can autonomously determine

whether it is a boundary node. In the second, boundary nodes can be clustered according to the coverage holes and the outer boundary of WSNs that they enclose.

3. Preliminary

3.1. Network Model and Assumption. Imagine the following scenario: a WSN is formed by n homogeneous sensor nodes v_i ($i = 1, \dots, n$) that are randomly deployed in a 2-dimensional monitored area \mathcal{A} . The WSN is modeled as a connected graph $G = (V, E)$, where V represents the set of all nodes in the WSN and E denotes the set of edges (communication links) between the pairwise nodes. In this model, it is assumed that each node has a unique ID and knows its own location via GPS or some other localization approaches [32, 33]. Besides, we assume that all the nodes can communicate directly with the adjacent nodes within the communication range and collect their own one- and two-hop neighbors. At the same time, this work assumes that the sensing area and communication range of node v_i in WSN are regular disks with R_s and R_c as the radii, respectively, centered at v_i . Here, R_s equals R_c .

3.2. Definitions

Definition 1 (neighbor set). Let $N_k(v_i)$, described in formula (1), denote the k -hop neighbor set of node v_i ; node v_u is called the k -hop neighbor of node v_i if $v_u \in N_k(v_i)$. Let $N_{1,2}(v_i)$, described as formula (2), represent the union set of one- and

two-hop neighbor nodes of v_i . Let $|N_{1,2}(v_i)|$ denote the node number of $N_{1,2}(v_i)$:

$$N_k(v_i) = \{v_u \mid (k-1) \times R_c < d_{v_u, v_i} \leq k \times R_c, (u \neq i, k = 1, 2, \dots, n)\} \quad (1)$$

$$N_{1,2}(v_i) = N_1(v_i) \cup N_2(v_i), \quad (2)$$

where d_{v_u, v_i} is the Euclidean distance between the nodes v_u and v_i .

Definition 2 (sensing area). The sensing area of node v_i , denoted as $S(v_i)$, is described as formula (3). $S(v_i)$ is a circular disk with R_s as the radius and v_i as the center. As shown in Figure 1, with the location of node v_i as the origin of local Cartesian Coordinate System (CCS), the circular disk enclosed by dashed line is the sensing area of node v_i ; the dashed line represents the boundary of sensing area, denoted as $\partial S(v_i)$:

$$S(v_i) : x^2 + y^2 \leq R_s^2. \quad (3)$$

Definition 3 (absolute angle). As shown in Figure 1, a local CCS is created with node $v_i(x_{v_i}, y_{v_i})$ as the origin, and $v_m \in N_{1,2}(v_i)$. The coordinate of v_m in local CCS is (x_{v_m}, y_{v_m}) . The absolute angle of v_m refers to the forward angle from x -axis positive direction to the ray $\overrightarrow{v_i v_m}$ and it is denoted by $\angle \text{abs}(v_i, v_m)$ ($\angle \text{abs}(v_i, v_m) \in [0^\circ, 360^\circ]$) or $\angle \text{abs}(v_i, k)$; k represents the position of v_m in $N_{1,2}(v_i)$ where the nodes are sorted according to the ascent of absolute angles of nodes:

$$\angle \text{abs}(v_i, v_m) = \begin{cases} \text{mod} \left(360 + \arctan \left(\frac{y_{v_m} - y_{v_i}}{x_{v_m} - x_{v_i}}, 360 \right) \right), & x_{v_m} \neq x_{v_i} \\ 90^\circ, & x_{v_m} = x_{v_i} \wedge y_{v_m} > 0 \\ 270^\circ, & x_{v_m} = x_{v_i} \wedge y_{v_m} < 0. \end{cases} \quad (4)$$

Definition 4 (central angle). As shown in Figure 1, an angle formed by the two rays, $\overrightarrow{v_i v_m}$ and $\overrightarrow{v_i v_n}$, is called a central angle, denoted as $\angle(v_m, v_i, v_n)$ ($\angle(v_m, v_i, v_n) \in [0, \pi]$).

Definition 5 (sensing sector). $S(v_i)$ can be divided into k ($k = 1, \dots, |N_{1,2}(v_i)|$) sectors by $N_{1,2}(v_i)$; each sector is called a sensing sector and the k th sensing sector is denoted by $\mathbb{S}(v_i, k)$; if the central angle corresponding to $\mathbb{S}(v_i, k)$ is $\angle(v_m, v_i, v_n)$, then $\mathbb{S}(v_i, k)$ can also be denoted by $\mathbb{S}(v_m, v_i, v_n)$. The sectors satisfy the following formula:

$$S(v_i) = \bigcup_{k=1}^{|N_{1,2}(v_i)|} \mathbb{S}(v_i, k). \quad (5)$$

Definition 6 (communication range). As shown in Figure 1, the communication range of node v_i is the circular disk

enclosed by dashed line, with v_i as its center and R_c as its radius; $R_c = R_s$ in this paper.

Definition 7 (investigated node (ItN)). ItN is the node that has already collected the neighbor information and is ready to perform the boundary node detection algorithm.

Definition 8 (boundary node (BN)). A node is called a BN if it lies on the border of holes or the outer boundary of WSNs. A set of boundary nodes is denoted by BNs. If a node is not a BN, then it is called a non-BN; the set of nonboundary nodes is denoted by non-BNs, as illustrated in Figure 2.

Definition 9 (coverage hole). In this paper, a coverage hole means a hole that is not covered by the sensing area of any node in \mathcal{A} , as illustrated in Figure 2.

A summary of the symbols is described in Symbol.

Input: v_i , Graph $G = (V, E)$
Output: H_{v_i}

- (1) Set $ItN \leftarrow v_i$ // choosing a node v_i randomly from set V as ItN
- (2) $H_{v_i} \leftarrow false$
- (3) Collect $N_1(v_i)$ and $N_2(v_i)$ of node v_i
- (4) Set $N_{1,2}(v_i) = N_1(v_i) \cup N_2(v_i)$
- (5) Calculate the $\angle abs(v_i, k)$ ($k = 1, \dots, |N_{1,2}(v_i)|$) of nodes in $N_{1,2}(v_i)$
- (6) **if** $(\angle abs(v_i, m) = \angle abs(v_i, l)) \wedge (d_{m,i} \leq \bar{d}_{l,i})$ **then**
- (7) $N_{1,2}(v_i) = N_{1,2}(v_i) - \{v_i\}$
- (8) Sort the nodes of $N_{1,2}(v_i)$ on the basis of their absolute angles in the ascending way
- (9) **if** the absolute angle of the node in $N_{1,2}(v_i)$ meets any one of the following conditions:
- (10) (a) $\angle abs(v_i, |N_{1,2}(v_i)|) \leq \pi$, (b) $\angle abs(v_i, 1) > \pi$, (c) $\forall k, \angle abs(v_i, k) \in (\pi/2, 3\pi/2]$,
 (d) $\forall k, \angle abs(v_i, k) \in [0, \pi/2] \mid \angle abs(v_i, k) \in (3\pi/2, 2\pi]$. **then**
- (11) $H_{v_i} \leftarrow true$
- (12) **else**
- (13) **for** $j \leftarrow 1$ **to** $|N_{1,2}(v_i)|$ **do**
- (14) **if** $j < |N_{1,2}(v_i)|$ **then**
- (15) $H_{v_i} \leftarrow H_{v_i} \cup sectorCover((N_{1,2}(v_i))_j, (N_{1,2}(v_i))_{j+1})$
 // $(N_{1,2}(v_i))_j$ and $(N_{1,2}(v_i))_{j+1}$ represent the elements of j and $j+1$ of $N_{1,2}(v_i)$, respectively
- (16) **else**
- (17) $H_{v_i} \leftarrow H_{v_i} \cup sectorCover((N_{1,2}(v_i))_{|N_{1,2}(v_i)|}, (N_{1,2}(v_i))_1)$
- (18) **return** H_{v_i}

ALGORITHM 1: DSCS algorithm.

Input: $start, end, v_i, N_{1,2}(v_i)$
Output: $flag$

- (1) $flag \leftarrow false$
- (2) **if** $\mathbb{S}(N_{1,2}(v_i)_{start}, v_i, N_{1,2}(v_i)_{end})$ is covered by the sensing area of the neighbor of v_i **then**
- (3) $flag \leftarrow false$
- (4) **else**
- (5) $flag \leftarrow true$
- (6) **return** $flag$

ALGORITHM 2: Function sectorCover.

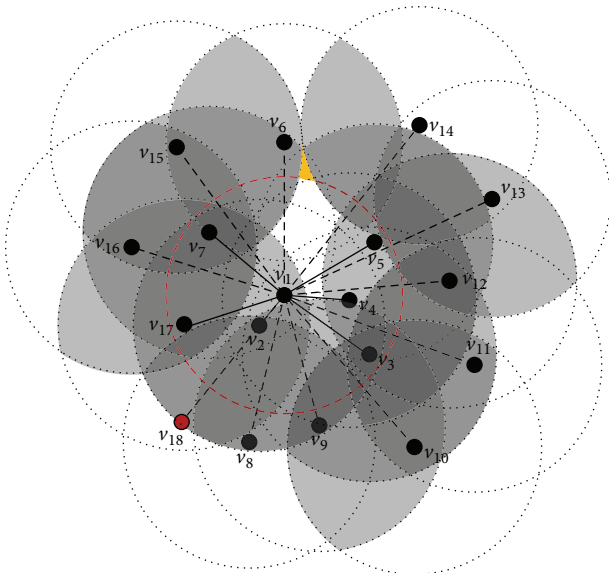


FIGURE 3: Illustration of BN detection.

inorder successor (denoted by v_{su}), and v_{su} , as the new v_c , is put into Q . Otherwise, change the scope of the absolute angle to $[0, \pi/2]$ and repeat the above procedure. If there does not exist v_{su} in $N_1(v_c)$, then repeat the same searching process in $N_2(v_c)$.

Step 3. Repeat Step 2 until v_c cannot find its v_{su} any longer; then change the scope of the absolute angle to $[\pi/2, \pi]$ and execute the same procedure as in Step 1. Note that the first v_{su} found in this procedure is called the first inflection node; after that, the searching procedure will be continued within the scope $[0, \pi]$.

Step 4. If v_{su} of v_c cannot be found in Step 3, change the scope of absolute angle to $[\pi, 3\pi/2]$ and continue the searching process as in Step 3. In this procedure, the first found node is marked as the second inflection node, and then the searching process is continued within the scope $[\pi/2, 3\pi/2]$.

Step 5. When v_{su} of v_c cannot be found in Step 4, change the scope of absolute angle to $[3\pi/2, 2\pi]$ and repeat the same

```

Input:  $G_{\text{BNs}} = (\text{BNs}, E(\text{BNs})), G, \delta$ 
Output: BoundCycle
(1)  $i \leftarrow 0$ 
(2) while  $\text{BNs} \neq \delta$  do
(3)    $Q \leftarrow \emptyset$ 
(4)    $i \leftarrow i + 1$ 
(5)    $v_{\text{st}} \leftarrow$  finding the starting node from BNs according to Step 2
(6)    $Q \leftarrow Q \cup \{v_{\text{st}}\}$ 
(7)   while the  $v_{\text{su}}$  of  $v_c$  is not equal to  $v_{\text{st}}$  or  $v_{\text{st}} \notin N_{1,2}(v_c)$  do
(8)      $v_{\text{su}} \leftarrow$  searching for  $v_{\text{su}}$  of  $v_c$  from BNs according to Steps 2–5
(9)      $Q \leftarrow Q \cup \{v_{\text{su}}\}$ 
(10)    examining the continuity according to Step 6
(11)     $\text{BoundCycle}[i] \leftarrow Q$ 
(12)     $\text{BNs} \leftarrow \text{BNs} \setminus \{\text{BoundCycle}[i], N_1(\text{BoundCycle}[i])\}$ 

```

ALGORITHM 3: DW algorithm.

searching procedure as in Step 4. If the first node found during the procedure is v_{st} or the one- or two-hop neighbor of v_{st} , then the searching procedure is done.

Each time the implementation from Step 1 to Step 5 is over, a boundary cycle formed by selected nodes is identified. Of all the boundary cycles, the first identified is the outer boundary, while the rest are coverage holes.

Step 6. Check the queue Q ; if there is any node in Q that has its two-hop neighbor as its v_{su} , insert a relay node that is the one-hop neighbor of both the node and its v_{su} . At the same time, the sum of the distance between the node and this relay node and the distance between the relay node and v_{su} should be the minimum.

Note that the relay node may be a non-BN.

Step 7. Remove all the nodes in Q and their one-hop neighbors from BNs.

Step 8. If $\text{BNs} = \emptyset$, repeat the procedure from Steps 1 to 7 until the set of BNs is empty or the number of the remaining nodes in the set is δ (in our practice $\delta = 2$).

We use Figure 2 to describe the whole procedure of DW. We set queue $Q = \emptyset$. The node v_{41} , bearing the minimum coordinate in all BNs, is selected as v_{st} according to Steps 1 and 2, being marked as v_c and pushed into Q ($Q = \{v_{41}\}$). Among the one-hop neighbors of v_c , v_{42} is the best choice to be v_{su} of v_c according to Step 2; then v_{42} takes the place of v_{41} to become v_c and is pushed into Q ($Q = \{v_{41}, v_{42}\}$). This process is executed repetitively until the first inflection point v_{24} appears. Next, DW changes the searching direction according to Step 3 and continues the same procedure. When the second inflection point v_{29} appears, DW changes the searching direction again according to Step 4 and repeats the same searching process until v_{su} of v_c is v_{st} or $v_{\text{st}} \in N_{1,2}(v_c)$; the searching process is over. In the following, DW finds that v_{su} of all nodes in Q are the one-hop neighbors of the nodes. Hence, the nodes enclosing the outer boundary are identified. Finally, DW removes all the nodes in Q ($Q = \{v_{41-44}, v_{21-40}\}$) and their one-hop neighbors $\{v_4, v_{9-20}\}$ from BNs. As a result, $\text{BNs} = \{v_{1-3}, v_5\}$ are left.

If DW finds $\text{BNs} \neq \emptyset$ after the outer boundary is identified, it resets $Q = \emptyset$, selects v_5 as v_{st} , and then performs the following operation. v_5 is marked as v_c and pushed into Q . Then, v_c finds v_3 as its v_{su} , being marked as new v_c and pushed into Q . At the same time, DW detects that v_c is the first inflection node according to Step 3; therefore, it changes the searching scope and finds v_2 as its v_{su} as well as the second inflection node according to Step 4, with v_2 being taken as v_c and pushed into Q . Here, DW changes the searching scope again and finds v_1 as v_{su} of v_c . Hence, DW takes v_1 as new v_c and pushes it into Q . Next, v_c finds that v_5 is both its one-hop neighbor and v_{st} according to Step 5. Therefore, DW begins to check v_{su} of the nodes in Q . As a result, DW finds that v_{su} of v_3 and v_{su} of v_2 are their two-hop neighbors, respectively, so v_4 is inserted between v_5 and v_3 according to Step 6. Similarly, v_8 is inserted between v_2 and v_3 . So far, all the nodes have their one-hop neighbors as their v_{su} in Q . That is to say, the nodes enclosing coverage holes are identified. In the following, DW removes all the clustered nodes in the second round of identification from BNs except v_4 and v_8 (v_4 and v_8 do not exist in BNs). At this moment, $\text{BNs} = \emptyset$, which means that the hole detection is over.

4.3. Theoretical Analysis. To prove the correctness of the implementation, we conduct the following theoretical analysis.

Lemma 10. Consider an ItN v_i , v_m , and v_n are any pair of adjacent neighbors ($v_m, v_n \in N_{1,2}(v_i)$), and $\mathfrak{S}(v_m, v_n) = S(v_m) \cap S(v_n)$, which is described as formula (7). If $\mathfrak{S}(v_m, v_n)$ is empty or $\mathfrak{S}(v_m, v_n)$ is not empty, the intersection of $\mathfrak{S}(v_m, v_n)$ with $\partial S(v_m, v_i, v_n)$ is empty. That is, $\mathfrak{S}(v_m, v_n) = \emptyset$ or $(\mathfrak{S}(v_m, v_n) \neq \emptyset) \wedge (\mathfrak{S}(v_m, v_n) \cap \partial S(v_m, v_i, v_n) = \emptyset)$; v_i is a BN:

$$\mathfrak{S}(v_m, v_n) = \begin{cases} (x - x_{v_m})^2 + (y - y_{v_m})^2 \leq R_s^2 \\ (x - x_{v_n})^2 + (y - y_{v_n})^2 \leq R_s^2, \end{cases} \quad (7)$$

where (x_{v_m}, y_{v_m}) and (x_{v_n}, y_{v_n}) are the coordinates of v_m and v_n , respectively, in the local CCS with v_i as the origin.

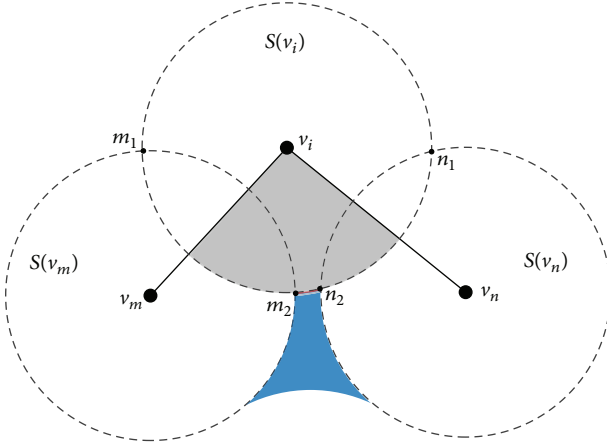


FIGURE 4: Illustration for Case 1.

Proof. Given a node v_i and any pair of adjacent neighbors v_m and v_n , we assume that v_m and v_n are located within the CCS with v_i as the origin. To prove the correctness of Lemma 10, we discuss it via the following three cases.

Case 1. If $\mathfrak{S}(v_m, v_n) = \emptyset$, $d_{v_m, v_n} > 2R_s$. Hence, $\partial\mathbb{S}(v_m, v_i, v_n)$ cannot be fully covered by $S(v_m)$ and $S(v_n)$; thereby, v_i is a BN according to Definition 8. As shown in Figure 4, the blue shaded area represents the coverage hole.

Case 2. In another scenario, $\mathfrak{S}(v_m, v_n) = \{P\}$, if point P lies on $\partial\mathbb{S}(v_m, v_i, v_n)$, $\partial\mathbb{S}(v_m, v_i, v_n)$ can be fully covered by $S(v_m)$ and $S(v_n)$. That is to say, there does not exist a hole in $\mathbb{S}(v_m, v_i, v_n)$ and v_i is a non-BN, as shown in Figure 5(a). Otherwise, no matter whether P is inside or outside $S(v_i)$, there must be a hole at least around it. Hence, v_i is a BN, as shown in Figures 5(b), 5(c), and 5(d).

Case 3. In the third case, $\mathfrak{S}(v_m, v_n) \neq \emptyset$; whether holes exist is determined by the position of $\mathfrak{S}(v_m, v_n)$, if $\mathfrak{S}(v_m, v_n) \cap \partial\mathbb{S}(v_m, v_i, v_n) = \emptyset$, as shown in Figures 6(a) and 6(b); there must exist a hole and v_i is a BN. Otherwise, no holes exist and v_i is a non-BN, as shown in Figures 6(c), 6(d), 6(e), and 6(f). \square

Lemma 11. *Considering an ItN v_i , $S(v_i)$ is divided into $|N_{1,2}(v_i)|$ by $N_{1,2}(v_i)$. If $\partial\mathbb{S}(v_i, k)$ ($k \in \{1, \dots, |N_{1,2}(v_i)|\}$) is not covered by the sensing areas of the neighbors of v_i , then v_i is a BN. Otherwise, it is a non-BN.*

Proof. According to Definition 5, $S(v_i)$ is divided into $|N_{1,2}(v_i)|$ sectors by the neighbors of v_i . If there exists a sector whose border cannot be covered by the sensing areas of the neighbors of v_i , according to Lemma 10 and Definition 8, node v_i is a BN. Otherwise, it is a non-BN. \square

Lemma 12. *Given a sensor node v_i , v_i is a BN if the absolute angle of a node in $N_{1,2}(v_i)$ meets any of the four cases: (a) $\angle\text{abs}(v_i, |N_{1,2}(v_i)|) \leq \pi$, (b) $\angle\text{abs}(v_i, 1) > \pi$, (c) $\forall k$,*

$\angle\text{abs}(v_i, k) \in (\pi/2, 3\pi/2]$, and (d) $\forall k$, $\angle\text{abs}(v_i, k) \in [0, \pi/2] \mid \angle\text{abs}(v_i, k) \in (3\pi/2, 2\pi]$.

Proof. We prove Case (a) and the other cases can be proved in the same way. As shown in Figure 7, nodes v_m and v_n are neighbors of node v_i . The three disk areas enclosed by the dashed lines are $S(v_m)$, $S(v_i)$, and $S(v_n)$, respectively. For v_m is the $|N_{1,2}(v_i)|$ neighbor of v_i , its absolute angle is the largest one. According to Case (a) of Lemma 12, $\angle\text{abs}(v_i, v_m) \leq \pi$. v_n is the adjacent node of v_m , and $v_n \in N_{1,2}(v_i)$; thus, $\angle\text{abs}(v_i, v_n) < \angle\text{abs}(v_i, v_m)$. The probable positions of v_m and v_n are shown in Figures 7(a), 7(b), and 7(c). In Figure 7(a), we assume that v_m , v_i , and v_n do not coincide with each other and they all lie on the x -axis and $\partial\mathbb{S}(v_m, v_i, v_n)$ can be completely covered by $S(v_m)$ and $S(v_n)$; meanwhile, $S(v_m) \cap S(v_n) \cap S(v_i) = \{P_1, P_2\}$. In addition, the radii of $S(v_m)$, $S(v_i)$, and $S(v_n)$ are equal. Hence, if $\partial\mathbb{S}(v_m, v_i, v_n)$ can be fully covered, $S(v_m)$, $S(v_i)$, and $S(v_n)$ must coincide; in other words, d_{v_m, P_2} , d_{v_i, P_2} , and d_{v_n, P_2} are equal. However, this contradicts with the assumption. Therefore, v_i is a BN when Case (a) is satisfied and all v_m , v_n , and v_i are on the x -axis. In Figures 7(b) and 7(c), we can clearly find that the blue shaded area cannot be covered by any sensors wherever v_m is located. Consequently, v_i is a BN when the position of $N_{1,2}(v_i)$ conforms to any case shown in Figures 7(b) and 7(c). \square

Lemma 13. *The BNs clustered by DW depict the outer boundary and coverage holes of WSNs, respectively.*

Proof. Because all the BNs located on the border of coverage holes and the outer boundary are identified by DSCS, we only need to cluster the BNs into different groups that map coverage holes and the outer boundary. Without loss of generality, within each group of the BNs, there must exist a BN with the minimum coordinate. Besides, the coverage holes lie within WSNs; hence, the BN with the minimum coordinate in all the BNs must lie on the border of the outer boundary. Therefore, the first identified boundary is the outer boundary. Similarly, the other boundary cycles identified in the subsequent rounds are coverage holes. \square

5. Experiments and Performance Evaluation

Both DSCS and DW are implemented in MATLAB 2013b on the platform of Windows 7. In the simulations, the sensor nodes are randomly deployed in a monitored square, the size of which is from $100\text{ m} \times 100\text{ m}$ to $500\text{ m} \times 500\text{ m}$. The number of the deployed nodes ranges from 200 to 2000 with the increment of 200. The communication radius of each node varies from 20 m to 50 m with the increment of 10 m. Besides, we compare our algorithms with the existing detection schemes, BRC and DHD, in terms of the correct detection rate, false detection rate, runtime, average energy dissipation, and algorithm complexity.

Figure 8 shows the simulation results in such a WSN that 2000 nodes are randomly deployed in a $500\text{ m} \times 500\text{ m}$ monitored area with coverage holes different in size, shape, and number. In Figure 8(c), the green points represent the BNs detected by DSCS and the purple represent the refined

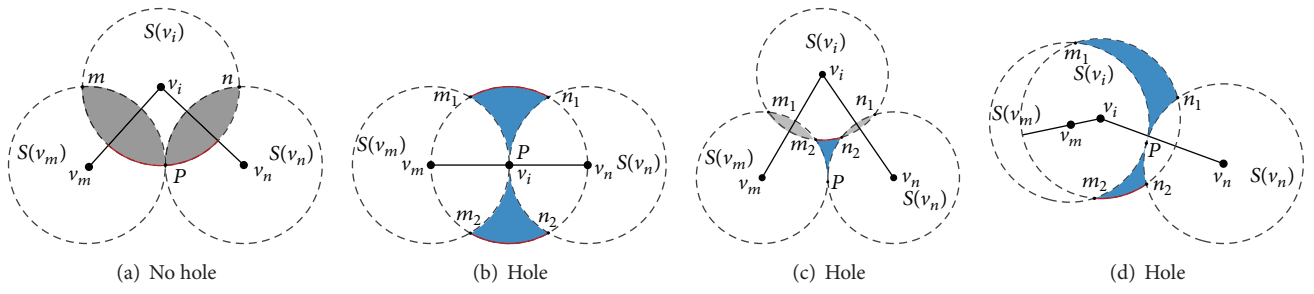


FIGURE 5: Illustration for Case 2.

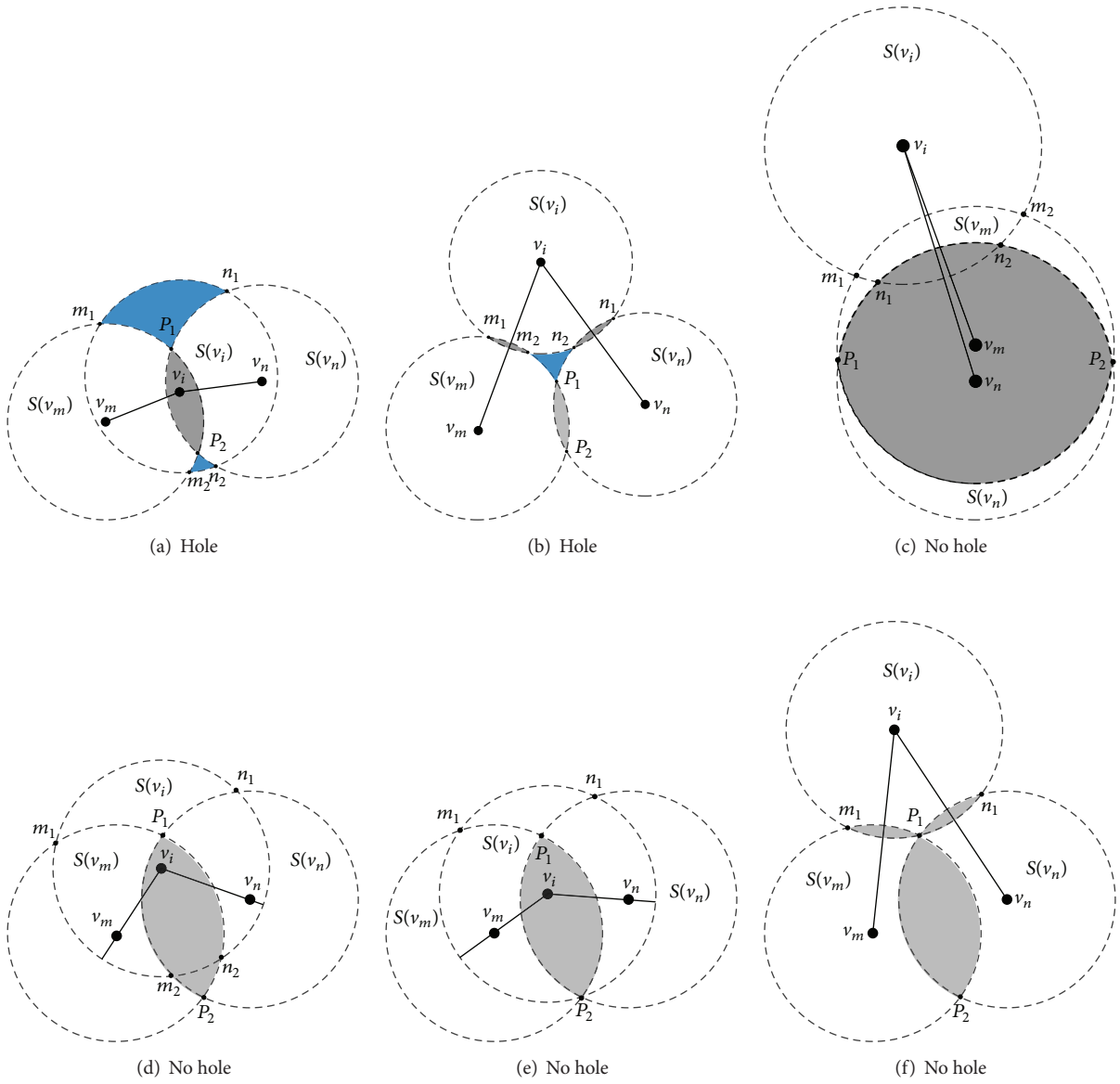


FIGURE 6: Illustration for Case 3.

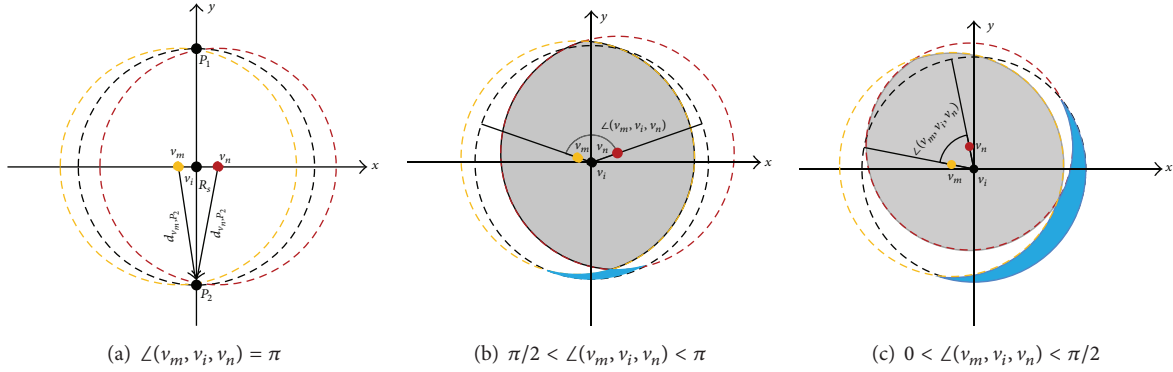


FIGURE 7: Illustration for Lemma 12.

ones by DW from the detected BNs. Moreover, DW connects the refined BNs to form different holes and the outer boundary of the WSN. Compared with BRC and DHD, DSCS hardly has any noise points.

Let BN_c denote the correctly detected BNs and BN_f ($BN_f = BNs \setminus BN_c$) denote the falsely detected BNs. The correct detection rate and false detection rate can be determined by the following formula:

$$\begin{aligned} \text{correctRate} &= \frac{|BN_c|}{|BNs|} \times 100\%, \\ \text{falseRate} &= \frac{|BN_f|}{|BNs|} \times 100\%. \end{aligned} \quad (8)$$

Figure 9 lists all the correct detection rates and false detection rates of algorithms DSCS, BRC, and DHD. For convenience, we apply the data set offered by Simek et al. [19] in this simulation. It can be observed that the correct detection rate of DSCS always keeps at more than 92% and the false detection rate keeps at less than 10% in the simulations with various sizes and different average node degrees. Although the correct detection rate increases and the false detection rate decreases along with the ascent of average node degree in all algorithms, it is clearly found that the fluctuation of the correct detection rate and false detection rate of BRC and DHD is significantly higher than that of DSCS, which shows that DSCS has a better robustness in all the simulation cases.

As shown in Figures 10(a) and 10(b), the number of BNs increases with the ascent of the size of the monitored area while it decreases with the ascent of the length of the communication radius. The reason is that when the same number of nodes is deployed in the monitored areas with different sizes, the node density decreases with the ascent of the size, which results in the reduction of the number of each node's neighbors and the enhancement of the BNs' occurrence probability. On the contrary, the node density increases with the ascent of the length of the communication radius, which lowers the BNs' occurrence probability. From Figure 10(c), we can clearly observe that the number of BNs detected by DSCS is neither the most nor the least, but

the outer boundary and the coverage holes are identified more accurately.

As for the runtime, Figures 11(a) and 11(b) show that it decreases with the enlargement of the monitored area and the reduction of the communication radius. The reason is that, given the number of nodes, the node density increases with the size reduction of the monitored area or the length increase of the communication radius. The increase of node density results in the fact that each node has to spend more time to transmit messages and compute its relation with the neighbors. As shown in Figure 11(c), although the runtime is increasing exponentially along with the ascent of the node number, the growth of runtime of DSCS and DW is slower than that of the other two algorithms.

As a key factor to prolonging the lifetime of WSNs, the energy efficiency during the phase of identifying BNs is evaluated through figuring out the energy each node consumes. During the simulations, all the algorithms apply the energy dissipation model as depicted in [34, 35].

Figure 12(a) shows the average energy dissipation of DSCS and DW in the WSNs with the size ranging from $100 \text{ m} \times 100 \text{ m}$ to $500 \text{ m} \times 500 \text{ m}$. It can be observed that the energy dissipation is inversely proportional to the size of the monitored area. The main reason is that each node needs to transmit more messages in the case of high node density. As mentioned in [34, 35], the dissipation of energy mainly concentrates on message transmission. Thus, the higher the node density is, the more the energy consumption is. Similarly, Figure 12(b) shows that the longer the communication radius is, the more the energy consumption is. In Figure 12(c), different algorithms show a common feature that the energy dissipation is proportional to the amount of nodes. However, the curves of DSCS and DW and BRC, especially of DSCS and DW, grow more smoothly than that of DHD. That is, DSCS is more energy-efficient than BRC and DHD. The same results are obtained from the comparison of remaining energy, as shown in Figure 12(d). The remaining energy of the nodes is the highest when DSCS and DW are employed.

To fully evaluate the performance of DSCS and DW, the computation complexity is considered and illustrated in Table 1. By the foregoing analysis, the implementation of DSCS and DW can be divided into five subprocedures,

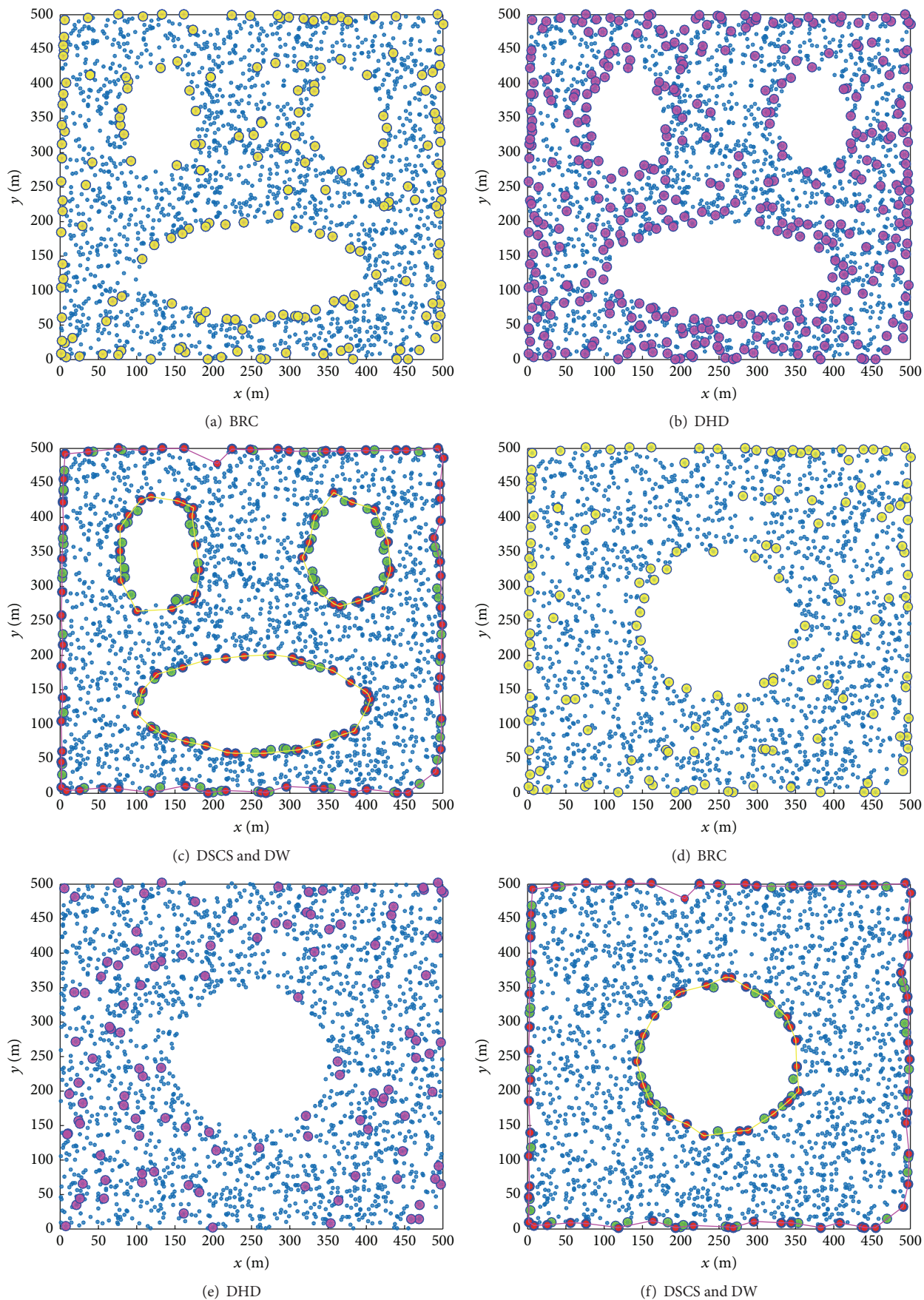


FIGURE 8: The comparison of detection.

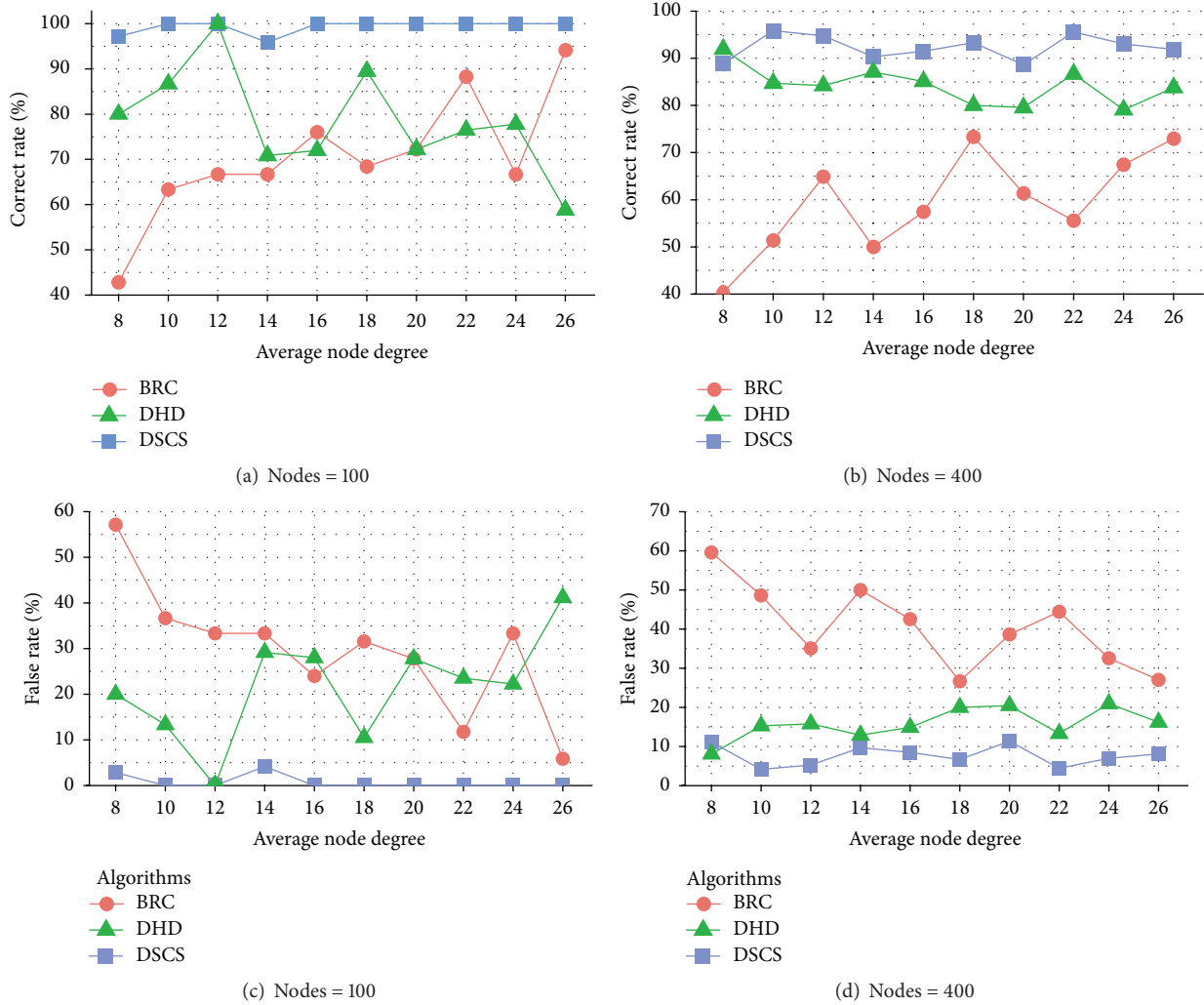


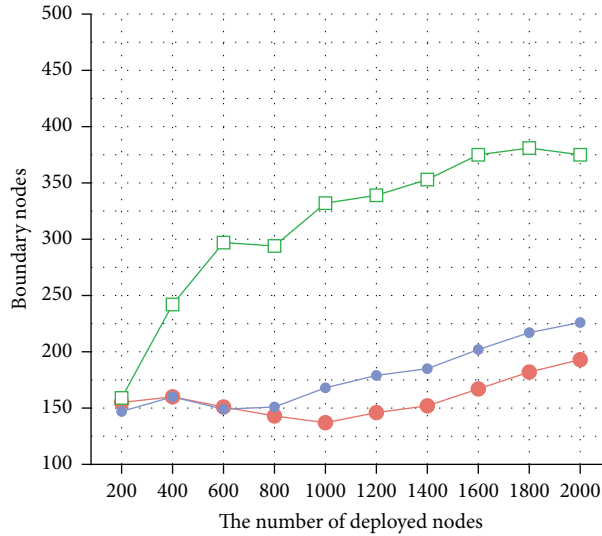
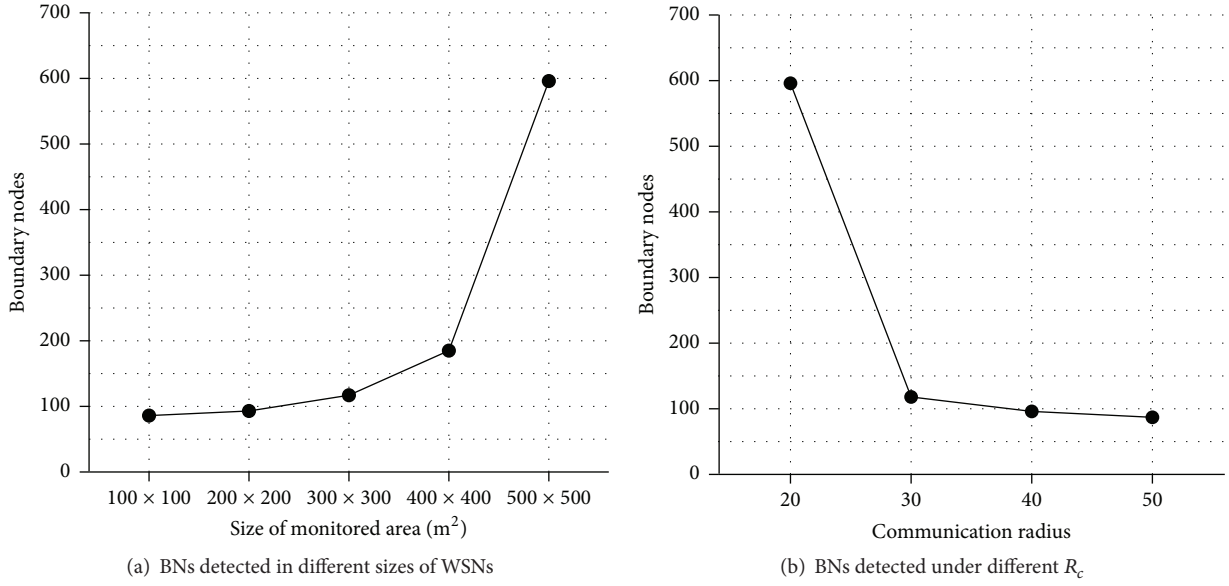
FIGURE 9: The comparison in terms of detection rate.

TABLE 1: The comparison in terms of algorithm complexity.

| Algorithm | Subprocedure | Complexity | Overall complexity |
|-------------|------------------------------------|----------------|--------------------|
| DSCS and DW | Collecting neighboring information | $O(n + m)$ | $O(34n + m)$ |
| | Computing absolute angle per node | $O(n)$ | |
| | Sorting nodes | $O(n \log(n))$ | |
| | Identifying BNs | $O(kn)$ | |
| | Connecting BNs that enclose holes | $O(a BNs)$ | |
| BRC | Choosing Good Neighbors | $O(n + m)$ | $O(60n + m)$ |
| | Identifying BNs | $O(3(k - 1)n)$ | |
| DHD | Finding max independent set | $O(n + m)$ | $O(86n + m)$ |
| | Creating isocontours | $O(I n)$ | |

namely, collecting neighbor information, computing absolute angle per node, sorting nodes, identifying BNs, and connecting them to enclose holes. Correspondingly, their complexities are $O(n + m)$, $O(n)$, $O(n \log(n))$, $O(kn)$, and $O(a(|BNs|))$, respectively. Here, $n = |V|$, $m = |E|$, $k = 21$

(k represents the average node degree of WSN), $a = 4$ (a denotes the number of coverage holes and the outer boundary of WSN), and $|BNs| = 226$. Therefore, the overall complexity of DSCS and DW is $O(34n + m + 904)$, approximately equal to $O(34n + m)$. The BRC is composed of choosing



(c) BNs detected by different algorithms

FIGURE 10: The comparison in terms of the number of detected BNs.

Good Neighbors and identifying BNs; their complexities are $O(n + m)$ and $O(3(k - 1)n)$, respectively. For $k = 21$, the overall computation complexity is $O(60n + m)$. The DHD algorithm can be divided into two steps. Firstly, it needs to find a max independent set I ; the complexity of the stage is $O(n + m)$. Secondly, every node of set I needs to serve as a seed to construct isocontours and the complexity of this stage is $O(|I|n)$ and $|I| = 85$. Hence, the overall computation complexity of DHD is $O(m + 86n)$. Obviously,

the complexity of DSCS and DW is lower than that of the other two algorithms.

6. Conclusions

In this paper, DSCS and DW are proposed for identifying boundary nodes and coverage holes in a WSN where nodes are randomly distributed. Besides, we simulate various scenarios and evaluate the proposed algorithms from different

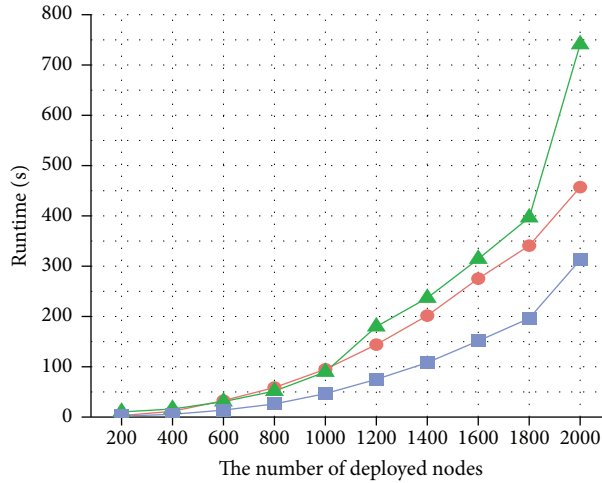
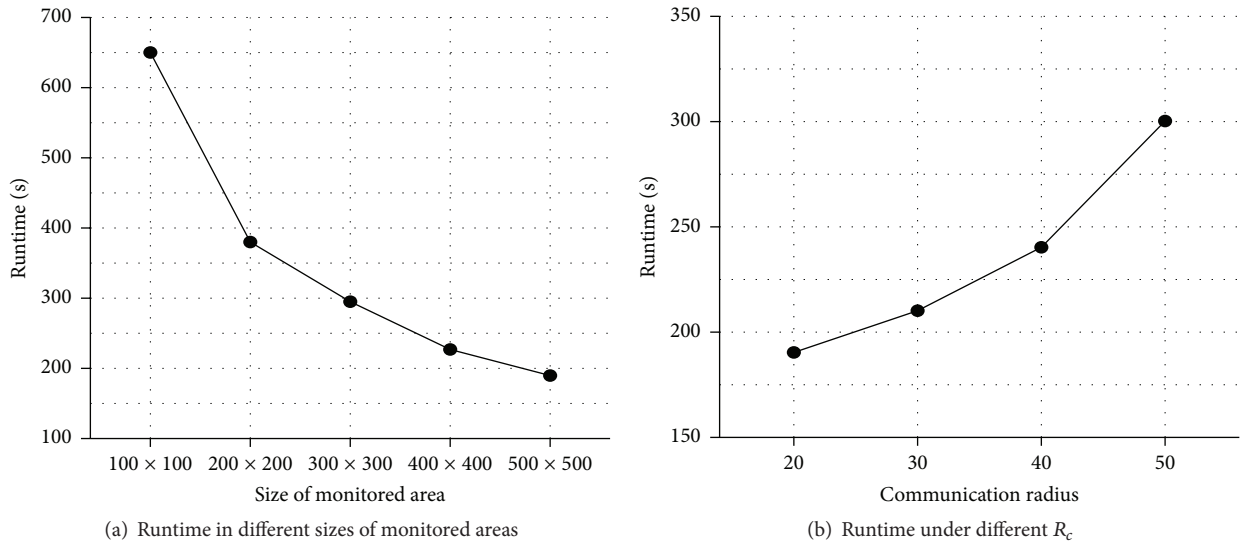


FIGURE 11: The comparison in terms of runtime.

aspects, including detection rate, runtime, energy efficiency, and algorithm complexity. In the performance evaluation, the effects from the size of WSN and the length of communication radius are taken into account. Simulation results show that DSCS bears the inspiring performance for detecting boundary nodes and DW works well in identifying coverage holes of different sizes and shapes. The algorithms developed in our work are effective in healing coverage holes, improving data forwarding efficiency, and prolonging the lifetime of WSN.

Symbol

- \mathcal{A} : The monitored area
- G : The connected graph that maps the WSN
- V : The set of nodes in the WSN
- E : The edge set (also called communication links) of G
- d_{v_i, v_j} : The Euclidean distance between nodes v_i and v_j
- $N_k(v_i)$: The k -hop neighbor set of node v_i and $k = 1, 2, \dots, n$

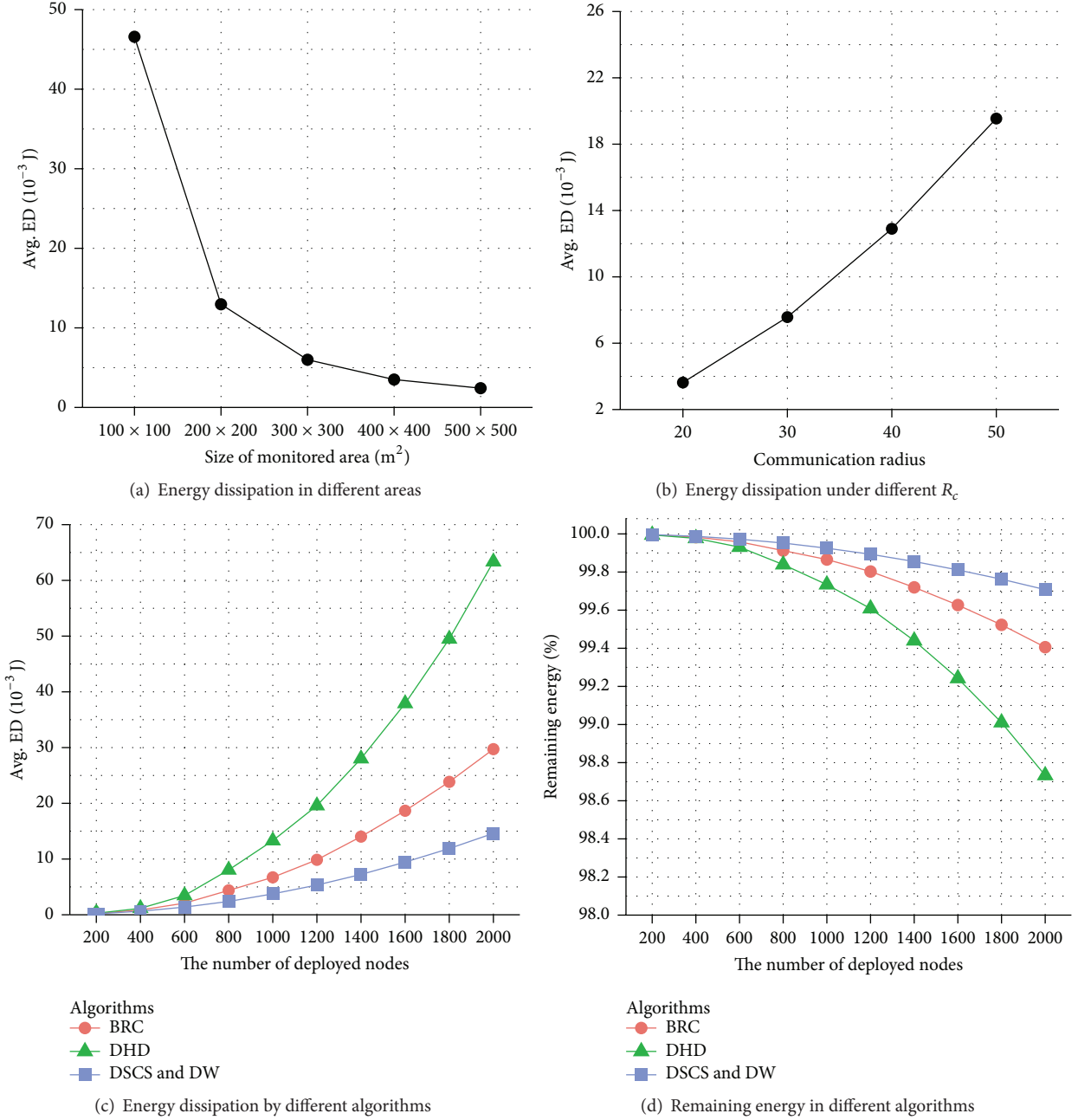


FIGURE 12: The comparison in terms of energy dissipation.

$N_{1,2}(v_i)$: The union set of one- and two-hop neighbors of node v_i

$\angle \text{abs}(v_i, v_m)$: The absolute angle of node v_m

$\angle(v_m, v_i, v_n)$: The central angle constructed by rays $\overrightarrow{v_i v_m}$ and $\overrightarrow{v_i v_n}$

$S(v_i)$: The sensing range of node v_i

$\partial S(v_i)$: The sensing border of node v_i

$\mathbb{S}(v_i, k)$: The sector of $S(v_i)$ and the number is k

$\mathbb{S}(v_m, v_i, v_k)$: The sector of $S(v_i)$ corresponding to $\angle(v_m, v_i, v_k)$

R_c : The communication radius of node

R_s : The sensing radius of node

BN: The boundary node of WSN

ItN: The investigated node of WSN.

Conflict of Interests

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

Acknowledgments

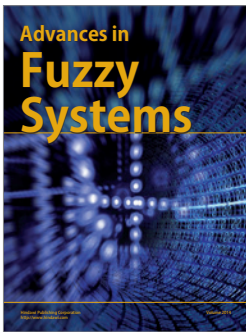
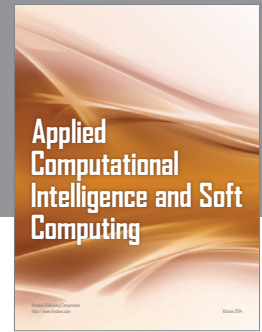
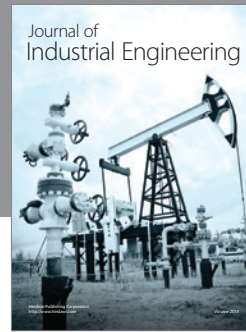
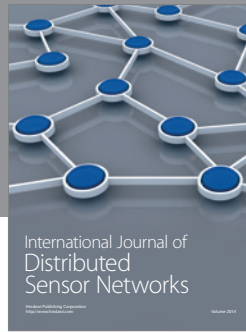
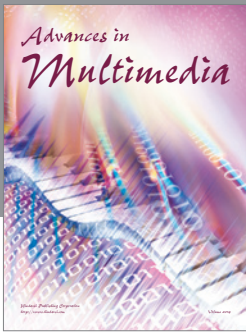
This work is supported by National Natural Science Foundation of China (NSFC) (51275491). The authors would like

to express sincere gratitude to Simek for his source code and data set, which offered them a great help in the WSN research.

References

- [1] T. Bokareva, W. Hu, S. Kanhere et al., "Wireless sensor networks for battlefield surveillance," in *Proceedings of the Land Warfare Conference*, pp. 1–8, Brisbane, Australia, October 2006.
- [2] B. Pannetier, J. Dezert, and G. Sella, "Multiple target tracking with wireless sensor network for ground battlefield surveillance," in *Proceedings of the 17th International Conference on Information Fusion (FUSION '14)*, pp. 1–8, IEEE, Salamanca, Spain, July 2014.
- [3] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 266–282, 2014.
- [4] P. K. Sahoo, J.-P. Sheu, and K.-Y. Hsieh, "Target tracking and boundary node selection algorithms of wireless sensor networks for internet services," *Information Sciences*, vol. 230, pp. 21–38, 2013.
- [5] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, "A robust, adaptive, solar-powered WSN framework for aquatic environmental monitoring," *IEEE Sensors Journal*, vol. 11, no. 1, pp. 45–55, 2011.
- [6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [7] J. M. L. P. Caldeira, J. J. P. C. Rodrigues, and P. Lorenz, "Toward ubiquitous mobility solutions for body sensor networks on healthcare," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 108–115, 2012.
- [8] S. Funke and C. Klein, "Hole detection or: how much geometry hides in connectivity?" in *Proceedings of the 22nd Annual Symposium on Computational Geometry (SCG '06)*, pp. 377–385, ACM, New York, NY, USA, 2006.
- [9] J. Vales-Alonso, F. J. Parrado-García, P. López-Matencio, J. J. Alcaraz, and F. J. González-Castaño, "On the optimal random deployment of wireless sensor networks in non-homogeneous scenarios," *Ad Hoc Networks*, vol. 11, no. 3, pp. 846–860, 2013.
- [10] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619–632, 2012.
- [11] S. Benkirane, A. Beni Hssane, M. Hasnaoui, M. D. El Ouadghiri, and M. Laghdir, "Performance analysis of routing protocols for mobile wireless sensor network in an environment with obstacles using nctuns tools," in *Proceedings of the Next Generation Networks and Services (NGNS '12)*, pp. 81–86, IEEE, Faro, Portugal, December 2012.
- [12] I. M. Khan, N. Jabeur, and S. Zeadally, "Hop-based approach for holes and boundary detection in wireless sensor networks," *IET Wireless Sensor Systems*, vol. 2, no. 4, pp. 328–337, 2012.
- [13] L. Shu, Y. Zhang, L. T. Yang, Y. Wang, M. Hauswirth, and N. Xiong, "TPGF: geographic routing in wireless multimedia sensor networks," *Telecommunication Systems*, vol. 44, no. 1-2, pp. 79–95, 2010.
- [14] A. Mostefaoui, M. Melkemi, and A. Boukerche, "Localized routing approach to bypass holes in wireless sensor networks," *IEEE Transactions on Computers*, vol. 63, no. 12, pp. 3053–3065, 2014.
- [15] M. R. Senouci, A. Mellouk, and K. Assnoute, "Localized movement-assisted sensor deployment algorithm for hole detection and healing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1267–1277, 2014.
- [16] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing holes in sensor networks," *Mobile Networks and Applications*, vol. 11, no. 2, pp. 187–200, 2006.
- [17] A. M. Khedr, W. Osamy, and D. P. Agrawal, "Perimeter discovery in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 69, no. 11, pp. 922–929, 2009.
- [18] J. S. Deogun, S. Das, H. S. Hamza, and S. Goddard, "An algorithm for boundary discovery in wireless sensor networks," in *High Performance Computing—HiPC 2005*, vol. 3769 of *Lecture Notes in Computer Science*, pp. 343–352, Springer, Berlin, Germany, 2005.
- [19] M. Simek, P. Moravek, D. Komosny, and M. Dusik, "Distributed recognition of reference nodes for wireless sensor network localization," *Radioengineering*, vol. 21, no. 1, pp. 89–98, 2012.
- [20] H.-C. Ma, P. Kumar Sahoo, and Y.-W. Chen, "Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1743–1756, 2011.
- [21] W.-T. Wang and K.-F. Ssu, "Obstacle detection and estimation in wireless sensor networks," *Computer Networks*, vol. 57, no. 4, pp. 858–868, 2013.
- [22] W. Li and W. Zhang, "Coverage hole and boundary nodes detection in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 48, pp. 35–43, 2015.
- [23] S. P. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann, "Neighborhood-based topology recognition in sensor networks," in *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop, ALGOSENSORS 2004, Turku, Finland, July 16, 2004. Proceedings*, vol. 3121, pp. 123–136, Springer, Berlin, Germany, 2004.
- [24] S. P. Fekete, M. Kaufmann, A. Kröller, and K. Lehmann, "A new approach for boundary recognition in geometric sensor networks," in *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG '05)*, pp. 82–85, Ontario, Canada, August 2005.
- [25] Y. Wang, J. Gao, and J. S. B. Mitchell, "Boundary recognition in sensor networks by topological methods," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 122–133, Los Angeles, Calif, USA, September 2006.
- [26] W.-C. Chu and K.-F. Ssu, "Location-free boundary detection in mobile wireless sensor networks with a distributed approach," *Computer Networks*, vol. 70, pp. 96–112, 2014.
- [27] B. Huang, W. Wu, G. Gao, and T. Zhang, "Recognizing boundaries in wireless sensor networks based on local connectivity information," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 897039, 12 pages, 2014.
- [28] A. Kröller, S. P. Fekete, D. Pfisterer, and S. Fischer, "Deterministic boundary recognition and topology extraction for large sensor networks," in *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA '06)*, pp. 1000–1009, ACM, Society for Industrial and Applied Mathematics, Miami, Fla, USA, January 2006.

- [29] O. Saukh, R. Sauter, M. Gauger, and P. J. Marrón, “On boundary recognition without location information in wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 6, no. 3, article 20, pp. 1–35, 2010.
- [30] R. Ghrist and A. Muhammad, “Coverage and hole-detection in sensor networks via homology,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 254–260, IEEE Press, Boise, Idaho, USA, April 2005.
- [31] X.-Y. Li, G. Calinescu, P.-J. Wan, and Y. Wang, “Localized delaunay triangulation with application in ad hoc wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 10, pp. 1035–1047, 2003.
- [32] Y. Xu, X. Gao, Z. Sun, and C. Li, “WSN node localization algorithm design based on RSSI technology,” in *Proceedings of the 5th IEEE International Conference on Intelligent Computation Technology and Automation (ICICTA '12)*, pp. 556–559, Zhangjiajie, China, January 2012.
- [33] H. Chen, G. Wang, Z. Wang, H.-C. So, and H. V. Poor, “Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 108–116, 2012.
- [34] G. Smaragdakis, I. Matta, A. Bestavros et al., “Sep: a stable election protocol for clustered heterogeneous wireless sensor networks,” in *Proceedings of the 2nd International Workshop on Sensor and Actor Network Protocols and Applications (SANPA '04)*, pp. 1–11, Boston, Mass, USA, August 2004.
- [35] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

