

## Research Article

# Boundary Detection Method for Large-Scale Coverage Holes in Wireless Sensor Network Based on Minimum Critical Threshold Constraint

Rong Jing,<sup>1</sup> Lingfu Kong,<sup>1</sup> and Liang Kong<sup>2</sup>

<sup>1</sup> School of Information Science and Engineering, Yanshan University, Hebei 066004, China

<sup>2</sup> School of Mathematics and Information Technology, Hebei Normal University of Science and Technology, Hebei 066004, China

Correspondence should be addressed to Rong Jing; [jingrong@ysu.edu.cn](mailto:jingrong@ysu.edu.cn)

Received 25 July 2014; Accepted 29 October 2014; Published 1 December 2014

Academic Editor: Yasuko Y. Maruo

Copyright © 2014 Rong Jing 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 existing coverage hole boundary detection methods cannot detect large-scale coverage hole boundary in wireless sensor network quickly and efficiently. Aiming at this problem, a boundary detection method for large-scale coverage holes in wireless sensor network based on minimum critical threshold constraint is proposed. Firstly, the optimization problem of minimum critical threshold is highlighted, and its formulaic description is constructed according to probabilistic sensing model. On the basis of this, the distributed gradient information is used to approximately solve the optimization problem. After that, local-scale rough boundary detection algorithm incorporating the minimum critical threshold and its iterative thinning algorithm are proposed according to blocking flow theory. The experimental results show that the proposed method has low computational complexity and network overhead when detecting large-scale coverage hole boundary in wireless sensor network.

## 1. Introduction

Wireless sensor network (WSN) is a network composed of a large number of sensor nodes by means of self-organization and multihop [1]. By acting as the interface between the logical information world and the objective physical world, WSN makes people obtain fine-detail surrounding environment information from microcosmic perspectives [2], so it is widely used in national defense, environmental monitoring, traffic management, medical treatment, antiterrorism, and so forth.

As one of the important measurement factors of QOS in WSN, coverage reflects the sensing service quality offered by WSN. Some nodes in WSN fail due to the limited node energy being exhausted or other accidental events, which trigger the phenomenon that sensing data will be lost, the conveyance time will be increased, and the topology may be separated, called coverage hole [3]. In general, the scale of coverage hole triggered by random node failure is small, so it can be recovered by WSN self-healing mechanism, such as waking sleep node [4], increasing the sensing power

[5], and adjusting the position of node [6]. However, the large-scale coverage hole is inevitable in the face of disaster environment monitoring, and its recovery is usually realized by external mobile device, like the mobile robot, the UAV, the unmanned airship, and so on, which has large sensing range, fast movement, and flexible deployment. Coverage hole boundary detection can be used for extracting further information about the structure and health of the network which is useful for routing, guiding, and management purposes [7]. It is the bridge for the external mobile device to obtain the large-scale coverage hole boundary information in WSN, and it directly affects repair efficiency of the external mobile device. As a result, it is important for designing a fast, accurate, and efficient large-scale coverage hole boundary detection method.

In recent years, the research of coverage hole boundary detection has become a new and hot topic in WSN. According to whether or not the sensor node needs location information, existing coverage hole boundary detection methods can be divided into two broad categories: geometry-based method and topology-based method, of which, according

to the relativity and absolutism of location information, geometry-based method can be further divided into absolute geometry method and relative geometry method. The absolute geometry method [8, 9] is based on the fact that the sensor node can accurately obtain the absolute location information, which can exactly discover coverage hole boundary in WSN, and the rates of missing coverage hole boundary are close to zero. However, the absolute geometry location information needs sensor nodes equipped with GPS, which will bring about big additional hardware overhead. By contrast, the relative geometry method [10–14] uses only programs developed by the algorithm to obtain self-location information in the relative coordinate system, which does not need additional hardware overhead. However, the deviation of location information will decrease with the network scale, which results in poor network scalability, and directly increases the computational complexity, rates of missing, and overhead of WSN. The topology-based method [15–18] uses topological properties related to WSN to recognize coverage hole boundary, which does not require any location information and has a better network scalability, computational complexity, and overhead. However, the topology-based method only detects coarse boundary of coverage hole and has a high missing rate of coverage hole boundary. In addition to their advantage and disadvantage, the detection object of both methods is the entire coverage holes in WSN, called undifferentiated boundary detection. Hence, for the coverage hole whose scale is specified, both methods have a high computational complexity and network overhead.

Based on minimum critical threshold, we propose a boundary detection method for large-scale coverage holes in WSN, that is, boundary detection for large-scale coverage holes in WSN based on minimum critical threshold constraint (BLW-MCT). The BLW-MCT belongs to geometric-topological hybrid method, whose characteristics are presented as topological global rough boundary detection and geometric local large-scale rough boundary detection and its iterative thinning. The main contributions of this paper are summarized as follows.

- (1) To reduce the computational complexity and network overhead of undifferentiated boundary detection, the minimum critical threshold optimization problem which can define large-scale coverage holes range in WSN is highlighted, and its formulaic definition and approximate solution are given.
- (2) To explore a better boundary detection method fusing minimum critical threshold, blocking flow based boundary detection transform thought is proposed, and local large-scale rough boundary detection algorithm and its iterative thinning algorithm realizing the transform thought are given.

Compared with the existing coverage hole boundary detection methods, the characteristics of the BLW-MCT are summarized as follows.

- (1) The existing topology-based coverage hole boundary detection methods usually require a certain amount

of border nodes as a priori knowledge, but the BLW-MCT does not limit it.

- (2) Contrary to existing topology-based methods, the scalability of existing geometry-based methods is poor, but the rates of missing coverage hole boundary are low. The BLW-MCT is geometric-topological hybrid method, which has not only good network scalability, but also low rates of missing coverage hole boundary.
- (3) Since both topology-based method and geometry-based method are undifferentiated boundary detection, the computational complexity and network overhead are high in the process of the large-scale coverage hole boundary detection. By contrast, the BLW-MCT uses minimum critical threshold to effectively limit the number of coverage holes in WSN, which, in part, decreases the computational complexity and network overhead.

## 2. Related Theory

In this section, the network model assumptions are given firstly. Then the probabilistic sensing model is described. Finally, blocking cut definition and its judgment method in blocking theory are introduced.

*2.1. Network Model Assumptions.* The assumptions of network model are as follows.

- (1) WSN is composed of  $n$  sensor nodes and a sink node randomly deployed in a two-dimensional square region  $A$ . WSN is event-driven network, and its sensor nodes are responsible for sending interesting events which occurred in  $A$  to the sink node. WSN is abstracted to a connected graph  $G(V, E)$ , where  $V = \{s_1, s_2, \dots, s_n\}$  is the set of sensor nodes in  $A$ , and  $E$  is the set of edges connecting sensor nodes.
- (2) The MAC layer of communication protocol of WSN adopts wake/sleep mode, in which each sensor node can autonomously switch between wake and sleep mode.
- (3) Each sensor node in WSN has the same maximum sensing radius  $R_s$  and communication radius  $R_c$ , and the sensing radius  $R_s$  can be autonomously adjusted.
- (4) All of the available sensors nodes are homogeneous in behavior of sensing, computing, communication, and other capabilities.
- (5) The location information of sensor node  $x$  can be obtained by existing positioning algorithms, and the sensor node in WSN can acquire the position of sink node from the flooding protocol.

*2.2. Probabilistic Sensing Model.* The probabilistic sensing model [19] reflects the signal strength decays with the distance from the source, which is close to the actual behavior of the sensor node, and more accurately describes the sensing

coverage ability of WSN. In the probabilistic sensing model, the probability that sensor node  $s_i$  in WSN detects the event which occurred at any position  $x$  in  $A$  is

$$p_i(x, s_i) = p_{0i}e^{-\lambda_i \|x - x_i\|}, \quad (1)$$

where  $p_{0i} \in (0, 1]$ ,  $\lambda_i$  is a positive constant, and  $\|x - x_i\|$  denotes the Euclidean distance between any position  $x$  in  $A$  and  $x_i$  of sensor node  $s_i$  in WSN. Furthermore, the sensor node  $s_i$  in  $p_i(x, s_i)$  has an omnidirectional sensing ability, and  $p_i(x, s_i)$  is a decreasing and derivative function.

**2.3. Blocking Flow Theory.** In blocking flow theory, flow units move in a certain direction in accordance with their own nature, leading to the network blocking at the position of some nodes, and then the back flow unit cannot move forward. Now the network traffic reaches saturation, resulting in quasi-blocking-cut (maximum blocking cut) [20].

The necessary and sufficient condition of judging whether a quasi-blocking-cut is a real blocking cut is as follows [21]: for any feasible flow in the network, if entrance arcs from the feasible flow and positive arcs in the quasi-blocking-cut all are saturated arcs, this quasi-blocking-cut is a real blocking cut, and the blocking cut is obtained by the feasible flow which is defined between positive arcs and opposite arcs located on network flow.

### 3. Optimization Problem of Minimum Critical Threshold

Before BLW-MCT execution, a simplified topology-based boundary detection method is used to pretreat WSN, obtaining global rough coverage hole boundary (hereinafter referred to as global rough boundary) and taking the global rough boundary as research object to carry out BLW-MCT research. In this section, the minimum critical threshold optimization problem is proposed. Then, its related symbol definitions are given, and the formulaic description of the optimization problem which is based on the symbol definition is established. Finally, an approximate solution for the optimization problem is given.

**3.1. Problem Posing.** Existing boundary detection method based on geometry or topology can detect all sizes of coverage hole in WSN, but they will bring about higher network overhead and computational complexity. Therefore, if the boundary detection methods can appropriately reduce the number of coverage holes being detected in WSN according to practical application demands, the computational complexity and network overhead of boundary detection will be dramatically reduced.

The research object in this paper is large-scale coverage hole, and the core goal is to design a fast, accurate, and efficient large-scale coverage hole boundary detection method. If the range of large-scale coverage holes in each global rough boundary can be quantized into a reasonable threshold, it will be easily introduced to BLW-MCT. That is an effective method to decrease the computational complexity

and network overhead of boundary detection method and improve detection precision.

In view of this, the concept of critical threshold which can exactly define the critical range between the large-scale coverage holes and other scales is introduced. Nevertheless, since the critical threshold is not existing isolated independently but is closely linked with network application, the critical threshold cannot be quantized into a constant. Considering the rates of missing coverage hole boundary under this condition, the critical threshold should be possibly small, so we can define the minimum value in critical threshold as minimum critical threshold and call the process of seeking the minimum critical threshold minimum critical threshold optimization problem.

**3.2. Symbol Definition.**  $B_1, \dots, B_m$  denotes  $m$  global rough boundaries in WSN, which are obtained by the step of weight calculations and point edge deletion in [15].

$G^k(V^k, E^k)$ ,  $k \in 1, \dots, m$ , denotes the network connected graph within any global rough boundary  $B_k$ , where  $V^k$  is the set of active sensor nodes within  $B_k$ , and  $E^k$  is the set of edges connecting sensor nodes.

$N^k = \{n_1^k, \dots, n_u^k\}$ ,  $k \in 1, \dots, m$ ,  $u \in 1, \dots, n$ , denotes the set of sensor nodes' ID in  $V^k$ .

$BN^k = \{bn_1^k, \dots, bn_v^k\}$ ,  $k \in 1, \dots, m$ ,  $v \in 1, \dots, n$ , denotes the set of sensor nodes' ID at  $B_k$ .

$\Omega_{n_i^k}^k$  denotes the set of all possible loop-free paths from sensor node  $s_{n_i^k}$  to  $s_{n_j^k}$  in  $V^k$ .

$\mathfrak{R}_k$  denotes the set of coverage sensor nodes within  $B_k$ , which is also called active sensor node in this paper.

$x_{i,k}$ ,  $i = 1, \dots, n$ ,  $k = 1, \dots, m$ , denotes a Boolean variable; if the sensor node  $s_i$  can simultaneously belong to  $V^k$  and  $\mathfrak{R}_k$ , then  $x_{i,k} = 1$ ; otherwise,  $x_{i,k} = 0$ .

$$EC^k = \{(n_i^k, n_j^k) : n_i^k, n_j^k \in N^k, n_i^k \neq n_j^k, c(s_{n_i^k}, s_{n_j^k}) = 1\}$$

denotes the set of strong links in  $E^k$ , where  $c$  is a Boolean variable, denoted by the following equation:

$$c(s_{n_i^k}, s_{n_j^k}) = \begin{cases} 1 & \|s_{n_i^k} - s_{n_j^k}\| \leq R_c \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

**3.3. Problem Formulation.** The coverage probability of sensing region can indirectly reflect the scale of coverage holes in WSN, so using the probabilistic sensing model to establish the coverage probability for any region in  $A$  can formulate minimum critical threshold optimization problem. In addition, the minimum critical threshold in different global rough boundary is different, but its formulation and solution are the same. Hence, this section considers the global rough boundary  $B_k$  as research object and gives its formulation description of minimum critical threshold optimization problem.

As the sensing region within the global rough boundary  $B_k$  can be simultaneously covered by multiple active sensor nodes located inside  $B_k$ , on the basis of the probabilistic sensing model from Formula (1), the collaborative detection

probability of any point  $x$  in  $B_k$  can be denoted by the following equation:

$$P_k(x, \mathfrak{R}_k) = 1 - \prod_{i=1}^n [1 - x_{i,k} \cdot p_i(x, s_i)]. \quad (3)$$

Since  $p_i(x, s_i)$  is a decreasing and derivative function, the collaborative detection probability of the sensing region within the global rough boundary  $B_k$  can be denoted by the following equation:

$$\mathfrak{S} = \int_{B_k} P_k(x, \mathfrak{R}_k) dx. \quad (4)$$

Based on Formula (4), the minimum critical threshold optimization problem of the global rough boundary  $B_k$  can be described as follows: while ensuring the connectivity between all active sensor nodes within  $B_k$ , the deviation between the individual detection probability of  $B_k$  and the collaborative detection probability of active sensor nodes within  $B_k$  is minimized. As an event occurring,  $B_k$ 's individual detection probability is equal to 1. Hence, minimum critical threshold optimization problem of the global rough boundary  $B_k$  can be denoted by the following equation:

$$\text{Objective function : } \quad \text{Min}_{\mathfrak{R}_k} \left[ 1 - \int_{B_k} P_k(x, \mathfrak{R}_k) dx \right], \quad (5)$$

$$\text{Subject to } \quad \sum_{i=1}^n x_{i,k} \leq 1, \quad \forall k = 1, \dots, m, \quad (6)$$

$$\Omega_{n_i}^k \neq \phi, \quad i \in 1, \dots, n, \quad (7)$$

where constraint (6) ensures that each sensor in the optimization result at most belongs to one global rough boundary. The constraint (7) ensures that active sensor nodes in the optimization result are all connected and interact with each other.

**3.4. Problem Approximate Solution.** The minimum critical threshold optimization problem from Formula (5) is that the position of all active sensor nodes in the global rough boundary  $B_k$  is viewed as optimization variable to seek the minimum critical threshold of  $B_k$ . It is obvious that this optimization problem is NP-hard. Hence, the object in this section is to give a distributed method solving this optimization problem, and each sensor node in the distributed method only uses own information to accomplish a quantitative calculation, which will effectively decrease the network overhead caused by transmitted message between sensor nodes.

The gradient information can provide a more accurate motion direction for sensor nodes in the optimization results, so the trajectory of sensor nodes within global rough boundary  $B_k$  optimized by the distributed gradient information can be denoted by the following equation:

$$s_i^{t+1} = s_i^t + \eta_t \left( \frac{\partial \mathfrak{S}}{\partial s_i} \right), \quad (8)$$

where  $\partial \mathfrak{S} / \partial s_i$  denotes the gradient of the sensor node  $s_i$ ,  $t$  denotes the iteration number, and  $\eta_t$  is the step size. In order to ensure the convergence of motion trajectory, the step series  $\{\eta_t\}$  is determined by [22].

Since  $p_i(x, s_i)$  is a continuous function,  $P_k(x, \mathfrak{R}_k)$  is also a continuous function of sensor node  $s_i$ . Hence, the gradient  $\partial \mathfrak{S} / \partial s_i$  can be denoted by the following equation:

$$\frac{\partial (-\mathfrak{S})}{\partial s_i} = - \int_{B_k} \frac{\partial P_k(x, \mathfrak{R}_k)}{\partial s_i} dx. \quad (9)$$

For Formula (9), the integral estimate has become the key to solving the gradient  $\partial \mathfrak{S} / \partial s_i$ , but it is very difficult to compute directly. This is because the solving process not only needs the information of global rough boundary  $B_k$  but also provides the position information of all active sensor nodes in  $B_k$ . To solve the problem, this paper introduces the discretization method. This is because the gradient of each sensor node can be calculated by the local information associated with this sensor node after the integral estimate component from Formula (9) is discretized. Thereupon, Formula (3) is brought into Formula (9) and performs discrete method. The gradient  $\partial \mathfrak{S} / \partial s_i$  is denoted by the following equation:

$$\frac{\partial \mathfrak{S}}{\partial s_i} = - \int_{\Omega_i} \prod_{k \in B_i} [1 - x_{i,k} \cdot p_i(x, s_i)] \frac{dp_i(x, s_i)}{ds_i} \frac{s_i - x}{d_i(x)} dx, \quad (10)$$

where  $B_i = \{k : \|s_i - s_k\| < 2R_s\}$ ,  $k = 1, \dots, n, k \neq i$ , denotes the set of sensor nodes associated with sensor node  $s_i$ ,  $d_i(x) \equiv \|x - s_i\|$ , and  $\Omega_i = \{x : d_i(x) \leq R_s\}$  denotes the sensing region of sensor node  $s_i$ .

From Formula (10), the gradient  $\partial P_k(x, \mathfrak{R}_k) / \partial s_i$  of sensor node  $s_i$  can be calculated by the local information associated with sensor node  $s_i$ . So bringing it into Formula (8), the optimal position set of active sensor node within  $B_k$  is denoted by the following equation:

$$OP_k = \{s_i^{t+1} : s_i^t \in \mathfrak{R}_k, i = 1, \dots, n, k = 1, \dots, m\}. \quad (11)$$

The optimal position set  $OP_k$  is used only as an estimated parameter to estimate the minimum critical threshold of  $B_k$  ideally, but in fact the sensor node  $s_i^t$  cannot be moved to optimal position  $s_i^{t+1}$ . Besides, for ease of designating the BLW-MCT in the next section, we give the formula for area of  $B_k$ 's minimum critical threshold, which is denoted by the following equation:

$$TH^{B_k} = \frac{\left[ \sum_{i=1}^v x_i^x (x_{i+1}^y - x_{i-1}^y) - \sum_{j=1}^w x_j^x (x_{j+1}^y - x_{j-1}^y) \right]}{2}, \quad (12)$$

where  $i \in BN^k$ ,  $j \in BN^{OP_k}$ ,  $BN^{OP_k} = \{bn_1^{OP_k}, \dots, bn_w^{OP_k}\}$ ,  $k \in 1, \dots, m$ ,  $w \in 1, \dots, n$ , denotes the ID set of sensor nodes located on boundary in  $OP^k$ ,  $x_i^x$  denotes the abscissa of sensor node  $s_i$ , and  $x_i^y$  denotes the ordinate of sensor node  $s_i$ .

#### 4. BLW-MCT

The minimum critical threshold of  $B_k$  is the optimal solution given by solving Formula (5) under ideal condition. However, it just might possibly reduce the number of coverage holes not suited to the critical range and cannot detect the large-scale coverage hole boundary in WSN. To seek the exact boundary of large-scale coverage hole, the transformation thought of boundary detection hospitable to define the scale of coverage hole is proposed based on blocking flow theory, and large-scale coverage hole coarse boundary (hereinafter referred to as local large-scale rough boundary, denoted by  $R_k$ ) detection algorithm fusing the minimum critical threshold  $Th^{B_k}$  and its iterative thinning algorithm are presented with the transformation thought.

**4.1. Issue Transformation.** Although using the minimum critical threshold to define the scale of coverage hole can effectively reduce the number of coverage holes processed by boundary detection method, the existing boundary detection method does not restrict the scale of coverage hole, which makes the minimum critical threshold not to be well-integrated into the existing boundary detection method. Hence, we banish the basic thinking of the existing boundary detection method and search for the resolution from the blocking cut definition and its judgment method in blocking theory. Since the blocking flow is with the directed graph as the research object, we begin with the directed process of WSN, that is to say, generating a directed shortest path tree which starts with the tree root. Then by comparing the coverage hole in the directed shortest path tree with the blocking cut, it is found that the coverage hole bifurcates at the node connected with positive arc, flows through the coverage hole in the opposite direction, and converges on the node connected with opposite arc. In the directed shortest path tree, the node connected with positive arc just is the least common ancestor of the directed shortest path tree and the node connected with opposite arc just is the leaf node of the directed shortest path tree, and the feasible flow hugging coverage hole is equivalent to the shortest path between the node connected with positive arc and the node connected with opposite arc, as shown in Figure 1.

According to the above analysis, the boundary detection problem can be translated into the following problem: finding a pair of leaf nodes with minimized sum of distance between the leaf nodes and their least common ancestor in the directed shortest path tree. After the boundary detection problem is transformed, the scale of coverage hole in WSN can be directly presented by the distance between leaf nodes and its least common ancestor, which is convenient for fusing the minimum critical threshold.

**4.2. Local Large-Scale Coarse Boundary Detection.** Based on the above boundary detection transforming idea, the related definitions of the local large-scale coarse boundary detection algorithm can be described as follows.

**Definition 1** (least common ancestor). For any two sensor nodes  $p, q$  in the directed shortest path tree, if there is a sensor

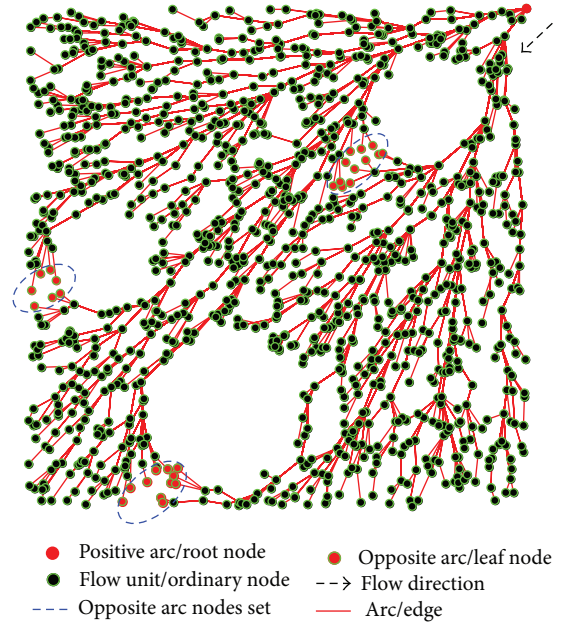


FIGURE 1: Coverage hole in directed shortest path tree.

node  $r$  which is far away from root node and is the ancestor of the  $p, q$ , the sensor node  $r$  is called the least common ancestor of sensor nodes  $p, q$ , denoted by  $LCA(p, q)$ .

**Definition 2** (cut node). The cut node is the sensor node which is the leaf node in the directed shortest path tree.

**Definition 3** (cut node set). The cut node set is the set of all cut nodes in the directed shortest path tree.

**Definition 4** (nonhomeomorphism). For any two manifolds in the mathematical field of topology, if a manifold is not transformed into the other manifold by a series of curved, extended, cut operation, the two manifolds are considered as nonhomeomorphism. Particularly for any two cut nodes  $p, q$  in the directed shortest path tree, if the path  $p$  to  $LCA(p, q)$  and the path  $q$  to  $LCA(p, q)$  are located on both sides of the coverage hole, two paths are nonhomeomorphism.

**Definition 5** (cut node pair). For any two sensor nodes  $p, q$  in the directed shortest path tree, if sensor nodes  $p, q$  are cut nodes having the same least common ancestor and path  $p$  to  $LCA(p, q)$  and the path  $q$  to  $LCA(p, q)$  are nonhomeomorphism, the sensor nodes  $p, q$  are called cut node pair, denoted by  $CNP(p, q)$ .

**Definition 6** (cut node pair set). The cut node pair set is the set of cut node pairs having the same least common ancestor, denoted by  $CNPS$ .

**Definition 7** (horizontal threshold). For any two sensor nodes  $p, q$  in the directed shortest path tree, if sensor nodes  $p, q$  are the cut node pair, the minimum value between the distance from  $p$  to  $LCA(p, q)$  and from  $q$  to  $LCA(p, q)$  is called horizontal threshold, denoted by  $HT$ .

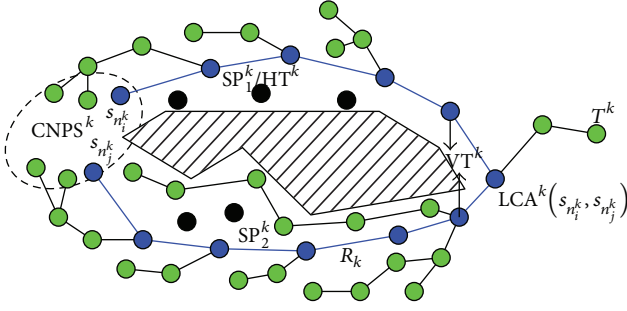


FIGURE 2: Diagram of local large-scale rough boundary detection in  $B^k$ .

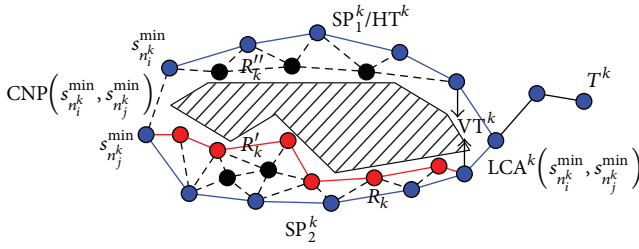


FIGURE 3: Topology graph of local large-scale rough boundary  $R^k$ .

**Definition 8** (vertical threshold). For any two sensor nodes  $p, q$  in the directed shortest path tree, if sensor nodes  $p, q$  are the cut node pair, the minimum distance from any sensor node in the path  $p$  to  $LCA(p, q)$  to any sensor node in the path  $q$  to  $LCA(p, q)$  is called vertical threshold, denoted by  $VT$ .

Based upon above definitions, the algorithm of local large-scale coarse boundary detection in  $B^k$  is described as follows. Firstly, the directed shortest path tree  $T^k$  is generated by the directed process of the network  $G^k(V^k, E^k)$  in  $B^k$ . Then, the cut node pair which keeps the sum of distance between the cut node pair and least common ancestor  $LCA^k(s_{n_i^k}, s_{n_j^k})$  minimum is found from the cut node pair set  $CNPS^k$  in  $T^k$ , and by comparing the  $Th^{B^k}$  with the product of the horizontal threshold  $HT^k$  and the vertical threshold  $VT^k$  of the cut node pair, the local large-scale coverage hole boundary  $R_k$  is generated (see Figure 2). The pseudocode of the local large-scale coarse boundary detection algorithm is as shown in Algorithm 1.

Algorithm 1 mainly includes three parts: the first part, from lines 1 to 21, is to generate the directed shortest path tree  $T^k$  of the network graph  $G^k(V^k, E^k)$  by the flooding routing algorithm. The second part, from lines 22 to 27, is to seek the minimum cut node pair  $CNP^k(s_{n_i^k}^min, s_{n_j^k}^min)$  from the  $CNPS^k$ . The third part, from lines 28 to 36, is to find local large-scale coarse boundary based on the minimum cut node pair  $CNP^k(s_{n_i^k}^min, s_{n_j^k}^min)$ .

**4.3. Iterative Thinning of Local Large-Scale Coarse Boundary.** For ease of describing the algorithm of iterative thinning for

local large-scale coarse boundary, the topology graph of local large-scale rough boundary  $R_k$  is shown in Figure 3.

Algorithm 1 can effectively find the local large-scale coarse boundary  $R_k$  (the blue solid lines in Figure 3), but there are some redundant shortest paths (the red solid lines in Figure 3) and the sleep sensor nodes (the black solid lines in Figure 3) inside  $R_k$ . As can be seen in Figure 3, the sensor nodes which are in the redundant shortest paths and sleep-state are closer to the actual large-scale coverage hole than the sensor nodes at the local large-scale rough boundary  $R_k$ . Hence, using the sensor nodes closer to the actual large-scale coverage hole in location to further thin the local large-scale coarse boundary  $R_k$  will make us obtain more accurate large-scale coverage hole boundary.

Although the rates of missing coverage hole boundary can be effectively reduced by waking the sleep nodes in  $R_k$ , the network overhead will rise accordingly. Hence, considering both the rates of missing coverage hole boundary and network overhead, the algorithm of iterative thinning for local large-scale coarse boundary is designed by the redundant shortest path information in  $R_k$ . The process of iterative thinning algorithm is divided into two stages: the first stage takes the network  $G^{k'}(V^{k'}, E^{k'})$  consisting of the local large-scale coarse boundary  $R_k$  and its internal redundant shortest path as the research object, and the boundary  $R'_k$  made up of three members, the blue-red path from  $s_{n_j^k}^min$  to  $LCA^k(s_{n_i^k}^min, s_{n_j^k}^min)$ , the blue path from  $s_{n_i^k}^min$  to  $LCA^k(s_{n_i^k}^min, s_{n_j^k}^min)$ , and the edge  $s_{n_i^k}^min s_{n_j^k}^min$  in Figure 3, is obtained by making use of the basic operations in graph theory. During this stage, the redundant shortest path information is used to isolate the sleep nodes between  $R_k$  and the redundant shortest path, which effectively decreases the network overhead. Similar to the iterative thinning steps of the first stage, the second stage will only replace the research object  $G^{k'}(V^{k'}, E^{k'})$  in the first stage with  $G^{k''}(V^{k''}, E^{k''})$  consisting of  $R'_k$  and its internal sleep nodes. After this stage, the accurate large-scale coverage hole boundary  $R''_k$  made up of three members, the blue-red path from  $s_{n_j^k}^min$  to  $LCA^k(s_{n_i^k}^min, s_{n_j^k}^min)$ , the black-blue path from  $s_{n_i^k}^min$  to  $LCA^k(s_{n_i^k}^min, s_{n_j^k}^min)$ , and the edge  $s_{n_i^k}^min s_{n_j^k}^min$  in Figure 3, is generated. The pseudocode of the algorithm of iterative thinning for local large-scale coarse boundary is as shown in Algorithm 2.

## 5. Experiment and Evaluation

In order to evaluate the performance of BLW-MCT, taking a small-scale WSN as an example, the execution process of BLW-MCT is given firstly. Then computational complexity of BLW-MCT is analyzed, and the computational complexity of each step in BLW-MCT is given. Finally, under different sensor node density and network scale, the proposed BLW-MCT is compared with the existing two classes of coverage hole boundary detection method by taking the number of packets, boundary detection time, and boundary discovery number as evaluation parameter.

**Input:** Global coarse boundary  $B_k$ , network graph  $G^k(V^k, E^k)$  inside  $B_k$ , the ID set  $N^k = \{n_1^k, \dots, n_v^k\}$  of all the sensor nodes in  $V^k$ , the minimum critical threshold of large-scale coverage hole  $Th^{B_k}$ .

**Output:** Local large-scale coarse boundary  $R_k$ .

- (1) **while**  $N^k$  **do**
- (2)   **for** each sensor node  $s_{n_i^k} \in V^k$  **do** //parallel process
- (3)      $t_{n_i^k}^k \leftarrow$  generating a random time remaining;
- (4)     **if**  $t_{n_i^k}^k == 0$  **do**
- (5)       generating a directed shortest path tree  $T^k(s_{n_i^k})$ ;
- (6)        $N^k \leftarrow N^k \setminus \{n_i^k\}$ ;
- (7)       broadcasting the ID  $n_i^k$  of  $s_{n_i^k}$  and its initial random time remaining  $t_{n_i^k}^k$ ;
- (8)     **end**
- (9)     **if** sensor node  $s_{n_j^k}$  receives the ID  $n_i^k$  sent by sensor node  $s_{n_i^k}$  **do**
- (10)       **if**  $s_{n_j^k}$  doesn't belong to any directed shortest path tree **do**
- (11)           $T^k(s_{n_i^k}) \leftarrow T^k(s_{n_i^k}) \cup s_{n_j^k}$ ;
- (12)           $N^k \leftarrow N^k \setminus \{n_j^k\}$ ;
- (13)       **else**
- (14)          **if**  $t_{n_i^k}^k \geq t_{n_j^k}^k$  **do**
- (15)            $N^k \leftarrow N^k \cup$  all node IDs in  $T^k(s_{n_i^k})$ ;
- (16)           broadcasting the ID  $n_j^k$  of  $s_{n_j^k}$  and its initial random time remaining  $t_{n_j^k}^k$ ;
- (17)          **end if**
- (18)       **end if**
- (19)     **end if**
- (20)   **end for**
- (21) **end while**
- (22)  $CNPS^k \leftarrow$  the cut node pair set in the directed shortest path tree  $T^k$ ;
- (23) **for** each cut node pair  $CNP^k(s_{n_i^k}, s_{n_j^k})$  in  $CNPS^k$  **do**
- (24)    $LCA^k(s_{n_i^k}, s_{n_j^k}) \leftarrow$  Tarjan ( $T^k, s_{n_i^k}, s_{n_j^k}$ ); //Tarjan is a offline least common ancestor algorithm
- (25)    $\Gamma \leftarrow$  computing the sum for path between  $s_{n_i^k}$  to  $LCA^k(s_{n_i^k}, s_{n_j^k})$  and  $s_{n_j^k}$  to  $LCA^k(s_{n_i^k}, s_{n_j^k})$ ;
- (26) **end for**
- (27)  $CNP^k(s_{n_i^k}^{\min}, s_{n_j^k}^{\min}) \leftarrow$  finding the cut node pair corresponding with the minimum value in  $\Gamma$ ;
- (28)  $LCA^k(s_{n_i^k}^{\min}, s_{n_j^k}^{\min}) \leftarrow$  Tarjan ( $T^k, s_{n_i^k}^{\min}, s_{n_j^k}^{\min}$ );
- (29)  $SP_1^k \leftarrow$  the shortest path between  $s_{n_i^k}^{\min}$  to  $LCA^k(s_{n_i^k}^{\min}, s_{n_j^k}^{\min})$ ;
- (30)  $SP_2^k \leftarrow$  the shortest path between  $s_{n_j^k}^{\min}$  to  $LCA^k(s_{n_i^k}^{\min}, s_{n_j^k}^{\min})$ ;
- (31)  $HT^k \leftarrow \text{Min}(SP_1^k, SP_2^k)$ ;
- (32)  $VT^k \leftarrow \text{Min}| SP_1^k - SP_2^k |$ ;
- (33) **if**  $HT^k \times VT^k \geq Th^{B_k}$  **do**
- (34)    $R_k \leftarrow SP_1^k \cup SP_2^k \cup (s_{n_i^k}^{\min}, s_{n_j^k}^{\min})$ ;
- (35) **end if**
- (36) **return**  $R_k$ ;

ALGORITHM 1: Local large-scale coarse boundary detection algorithm.

*5.1. Execution Process of BLW-MCT.* Figure 4(a) shows the initial distribution of WSN consisting of 1200 sensor nodes deployed randomly within a sensing region of 100 m  $\times$  100 m, where the sensor node is presented by the circle whose periphery is green and interior is black. The sensing radius of each sensor node is 5 m, and the communication radius of each sensor node is 7 m. The communication link is presented by blue line. The static sink node located in the center of

sensing region is presented by circle whose periphery is green and interior is red.

Based on the coverage degree of WSN, the contour plane related to the coverage degree is sketched in Figure 4(b). The coverage condition of initial WSN is intuitively read by the color bar located in the right of Figure 4(b); for example, the value zero corresponding with the deep blue indicates that the region colored by deep blue is not covered by any sensor node;

**Input:** The local large-scale coarse boundary  $R_k$ , the minimum cut node pair  $\text{CNP}^k(s_{n_i}^{\min}, s_{n_j}^{\min})$ , the least common ancestor  $\text{LCA}^k(s_{n_i}^{\min}, s_{n_j}^{\min})$ .

**Output:** The accurate large-scale coverage hole boundary  $R_k''$ .

- (1)  $G^{k'}(V^{k'}, E^{k'}) \leftarrow$  the network consisting of  $R_k$  and its internal redundant shortest paths; //The first stage
- (2)  $\text{SPT}^{k'} \leftarrow$  generating the shortest path tree of the  $G^{k'}(V^{k'}, E^{k'})$ ;
- (3)  $\text{SP}_1^{k'} \leftarrow$  searching the shortest path from  $s_{n_i}^{\min}$  to  $\text{LCA}^k(s_{n_i}^{\min}, s_{n_j}^{\min})$  in the  $\text{SPT}^{k'}$ ;
- (4)  $\text{SP}_2^{k'} \leftarrow$  searching the shortest path from  $s_{n_j}^{\min}$  to  $\text{LCA}^k(s_{n_i}^{\min}, s_{n_j}^{\min})$  in the  $\text{SPT}^{k'}$ ;
- (5)  $R_k' \leftarrow \text{SP}_1^{k'} \cup \text{SP}_2^{k'} \cup (s_{n_i}^{\min}, s_{n_j}^{\min})$ ;
- (6)  $G^{k''}(V^{k''}, E^{k''}) \leftarrow$  the network consisting of  $R_k'$  and its internal sleep nodes; //The second stage
- (7)  $\text{SPT}^{k''} \leftarrow$  generating the shortest path tree of the  $G^{k''}(V^{k''}, E^{k''})$ ;
- (8)  $\text{SP}_1^{k''} \leftarrow$  searching the shortest path from  $s_{n_i}^{\min}$  to  $\text{LCA}^k(s_{n_i}^{\min}, s_{n_j}^{\min})$  in the  $\text{SPT}^{k''}$ ;
- (9)  $\text{SP}_2^{k''} \leftarrow$  searching the shortest path from  $s_{n_j}^{\min}$  to  $\text{LCA}^k(s_{n_i}^{\min}, s_{n_j}^{\min})$  in the  $\text{SPT}^{k''}$ ;
- (10)  $R_k'' \leftarrow \text{SP}_1^{k''} \cup \text{SP}_2^{k''} \cup (s_{n_i}^{\min}, s_{n_j}^{\min})$ ;
- (11) **return**  $R_k''$ ;

ALGORITHM 2: Iterative thinning algorithm for local large-scale coarse boundary.

that is to say, the deep blue regions are the coverage holes of initial WSN.

Figure 4(c) gives the global coarse boundary of initial WSN, which is drawn by the PLEX (<http://mii.stanford.edu/research/comptop/programs/>) according to the weight calculations and point edge deletion mechanism in [15]. The global coarse boundary is presented by the closed polygon consisting of the red line.

Figures 4(d), 4(e), and 4(f) are key execution process of BLW-MCT. Taking any global coarse boundary  $B_k$  in Figure 4(c) as an example, the directed shortest path tree  $T^k$  of  $G^k(V^k, E^k)$  consisting of the sensor nodes in  $B_k$  is given by Figure 4(d), whose root node is presented by circle whose periphery is green and interior is red and shortest path is presented by red line. On the basis of  $T^k$  in Figure 4(d), the local large-scale coarse boundary  $R_k$  is given by Figure 4(e) according to the minimum critical threshold  $Th^{B_k}$ , whose boundary is presented by the closed polygon consisting of the blue line and node is presented by circle whose periphery is blue and interior is black. Based on the local large-scale coarse boundary  $R_k$ , the accurate large-scale coverage hole boundary  $R_k''$  is given by the two-stage iterative thinning, whose boundary is presented by the closed polygon consisting of the green line and node is made up of three members, the sleep node presented by circle whose interior is gray,  $R_k$ 's boundary node presented by circle whose periphery is blue and interior is black, and  $B_k$ 's boundary node presented by circle whose periphery is green and interior is gray.

**5.2. Computational Complexity Analysis for BLW-MCT.** The computational complexity of each step in BLW-MCT is shown in Table 1. The computational complexity analysis for each step in BLW-MCT is given as follows.

TABLE 1: Computational complexity of each step in BLW-MCT.

Step	Complexity
Global coarse boundary detection	$O(n^2)$
Minimum critical threshold approximate solution	$O(1)$
Local large-scale coarse boundary detection	$O(n^2)$
Local large-scale coarse boundary iterative thinning	$O(n^2)$

Similar to [15], the step of global coarse boundary detection just removes the detection process of edges and nodes related to boundary. Hence, the computational complexity of this step in the worst case is  $O(n^2)$ .

In order to reduce the network overhead from transferring message between sensor nodes, the step of minimum critical threshold approximate solution only uses the local information to solve the minimum critical threshold, so the computational complexity of this step is  $O(1)$ .

The step of local large-scale coarse boundary detection includes generating the directed shortest path tree, searching for the least common ancestor, obtaining the cut node pair set, and generating the local large-scale coarse boundary. The substep of generating the directed shortest path tree just needs each sensor to broadcast some message using the flooding routing algorithm, whose computational complexity in the worst case is  $O(n^2)$ . The substep of searching for the least common ancestor adopts the offline Tarjan algorithm to find the least common ancestor, so that the computational complexity of this substep in the worst case is  $O(m+Q)$ , where  $m$  is the number of sensor nodes in the directed shortest path tree and  $Q$  is the number of inquiries. Obviously,  $m$  and  $Q$  are far less than  $n$ , so that the computational complexity of this substep in the worst case is  $O(n)$ . The substep of obtaining the cut node pair set is finding the set of any two leaf nodes in



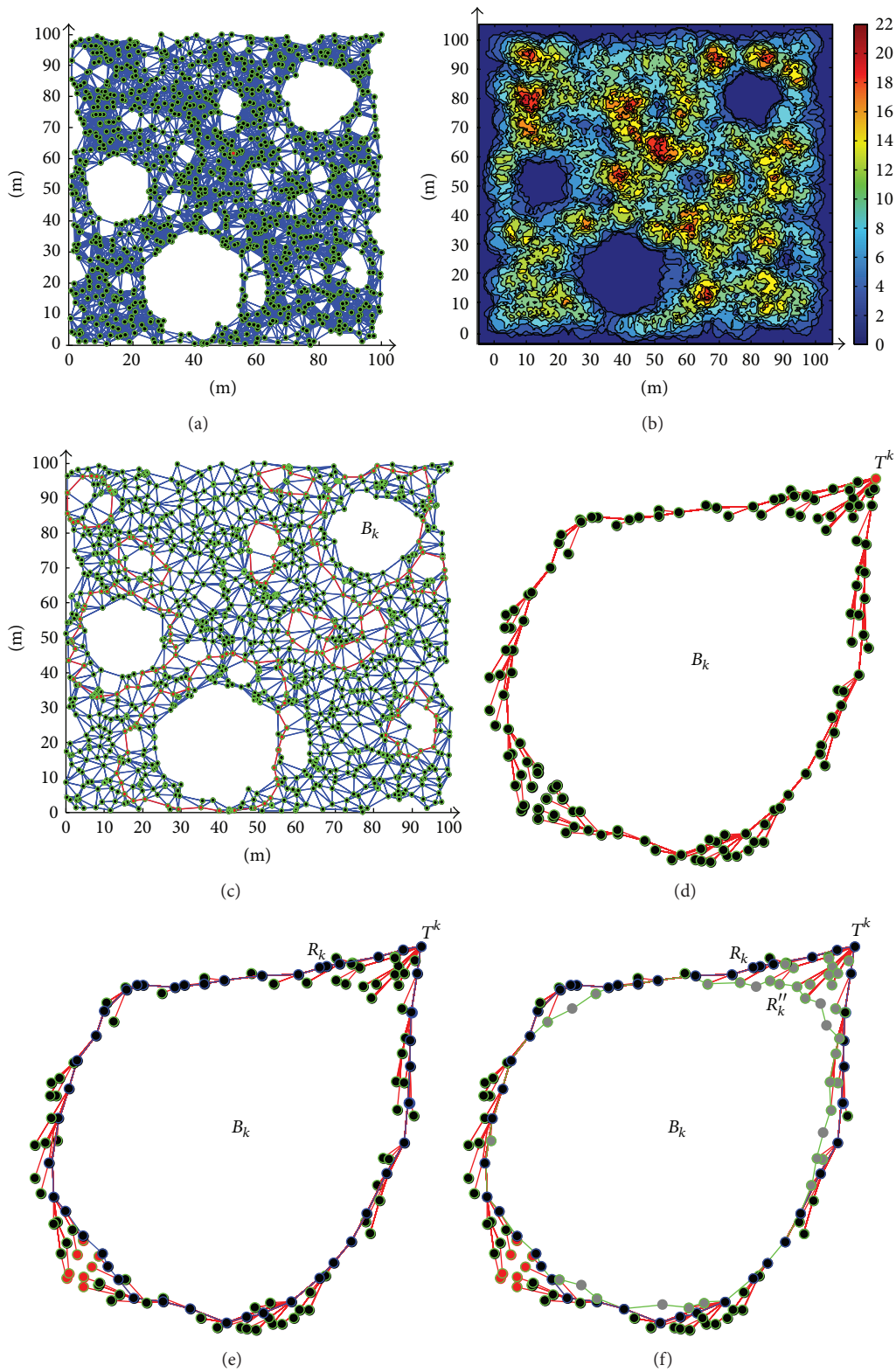


FIGURE 4: Implementation process of BLW-MCT: (a) initial distribution of WSN; (b) contour plane of initial WSN's coverage degree; (c) global coarse boundary of initial WSN; (d) shortest path tree in any global coarse boundary  $B_k$ ; (e) local large-scale coarse boundary  $R_k$ ; (f) accurate large-scale coverage hole boundary  $R''_k$ .

compliance with Definition 5. Obviously, the number of leaf nodes in the directed shortest path tree in the worst case is  $(m-1)$  and the number of judging whether any two leaf nodes comply with Definition 5 is  $\lceil (m-1)/2 \rceil^2$ , so the computational complexity of this substep in the worst case is  $O(n^2)$ . The substep of generating the local large-scale coarse boundary is searching for the cut node pair in the cut node pair set which makes the sum of distance from itself to the least common ancestor minimum, and the local large-scale coarse boundary consists of the path from the minimum cut node pair to the least common ancestor. In the worst case, the computational complexity of this substep is  $O(n)$ . According to the above computational complexity of each substep, the computational complexity of local large-scale coarse boundary detection in the worst case is  $O(n^2)$ .

The step of local large-scale coarse boundary iterative thinning mainly includes two processes of generating the shortest path tree: the shortest path tree in the first process is generated from the network graph consisting of the local large-scale coarse boundary and its internal active nodes, whose computational complexity is  $O(p^2)$ , where  $p$  is the number of active nodes in the local large-scale coarse boundary. Obviously,  $p$  is less than  $m$  and  $m$  is far less than  $n$ , so that the computational complexity in the first process in the worst case is  $O(n^2)$ ; the shortest path tree in the second process is generated from the network graph consisting of the first iterative thinning boundary and its internal sleep nodes, whose computational complexity is  $O(q^2)$ , where  $q$  is the number of sleep nodes in the first iterative thinning boundary. Obviously,  $q$  is less than  $m$  and  $m$  is far less than  $n$ , so that the computational complexity of generating this shortest path trees in the worst case is  $O(n^2)$  too. Hence, according to the computational complexity of above two processes of generating the shortest path tree, the computational complexity of local large-scale coarse boundary iterative thinning in the worst case is  $O(n^2)$ .

Since BLW-MCT is a distributed method, the computational complexity of BLW-MCT is equivalent to the computational complexity of any global coarse boundary  $B_k$  in the worst case. From the above analysis, the computational complexity of BLW-MCT is  $O(n^2)$ .

**5.3. Comparative Experiment.** To evaluate the performance of BLW-MCT, we compare it with the existing two classes of coverage hole boundary detection method, which are, respectively, localization-based boundary detection method [14] (LBM) and homology-based boundary detection method [18] (HBM). Besides, we introduce the size of packets, boundary detection time, and boundary discovery number as evaluation parameter evaluating the three methods mentioned above. In those three evaluation parameters, the size of packets is the sum of packets demanded by the process that boundary detection method discovers all coverage holes in WSN, which can adequately reflect the network overhead. The time during boundary detection is from the detection with the first sensor node until all coverage holes in WSN are detected, which can effectively measure the computational complexity of the boundary detection method. The boundary

discovery number is the number of coverage hole boundaries detected by the boundary detection method, which can reflect the accuracy of the boundary detection method to some extent.

To accomplish the comparative experiment in this section, a square sensing region of  $M \times M$  is established, where  $n$  sensor nodes are deployed randomly. The sink node is located in the center of sensing region and its position is fixed. The sensing radius of all sensor nodes is 7 m, and its communication radius is 10 m. Considering the randomness of network topology, we repeat 50 experiments for the size of packets and the boundary detection time, respectively, and select the average of 50 experiments as the results. For the boundary discovery number, if selecting the average of 50 experiments as the results will produce large error, this is because the large-scale coverage hole is only discussed in this paper, and the boundary discovery number is relatively small. So the sum of 50 experiments is selected as result of the boundary discovery number.

Figure 5 shows the performance comparison of BLW-MCT, LBM, and HBM in terms of the size of packets, boundary detection time, and boundary discovery number with the increase of  $n$  in WSN. During the experiment, sensing region is always  $100 \text{ m} \times 100 \text{ m}$ . As shown in Figure 5, the sizes of packets, boundary detection time, and boundary discovery number of the three methods mentioned above all gradually decrease with increasing the number of sensor nodes  $n$ .

In Figure 5(a), the sizes of packets of BLW-MCT, LBM, and HBM all gradually decrease with the increase of  $n$ . This is because the increasing of  $n$  brings about the decrease of the coverage hole number under unchanged sensing region; then the sizes of packets generated by BLW-MCT, LBM, and HBM all decrease. However, the decreased degrees of the size of packets are different, in which the size of packets generated by BLW-MCT is minimum, HBM is about 10% higher than BLW-MCT, and LBM is about 15% higher than BLW-MCT. This is because the size of packets of BLW-MCT only comes from the large-scale coverage holes in WSN detected by BLW-MCT, but LBM and HBM belong to the undifferentiated boundary detection, whose packets come from all coverage holes in WSN detected by them. Hence, the sizes of packets of LBM and HBM are greater than BLW-MCT. Besides, there is the triangle coverage hole in our experiment, but this type of coverage hole in WSN is not detected by HBM. So the size of packets of HBM is 5% lower than LBM. In Figure 5(b), the boundary detection time of BLW-MCT is less than LBM and HBM, and the time difference between them will decrease with the increasing of  $n$ . This is because the sparser the sensor node density in WSN, the higher the appearance probability of the large-scale coverage hole. We can also see from the figure that the boundary detection times of LBM and HBM decrease slowly in the beginning and then gradually increase. This is because the increase of  $n$  does not decrease the number of coverage holes in WSN in the beginning, and the number of coverage holes in WSN rapidly decreases with  $n$  further increase. In Figure 5(c), the T type of line in Figure 5(c) presents the total number of real coverage hole boundaries in 50 experiments, and the remaining histograms are the

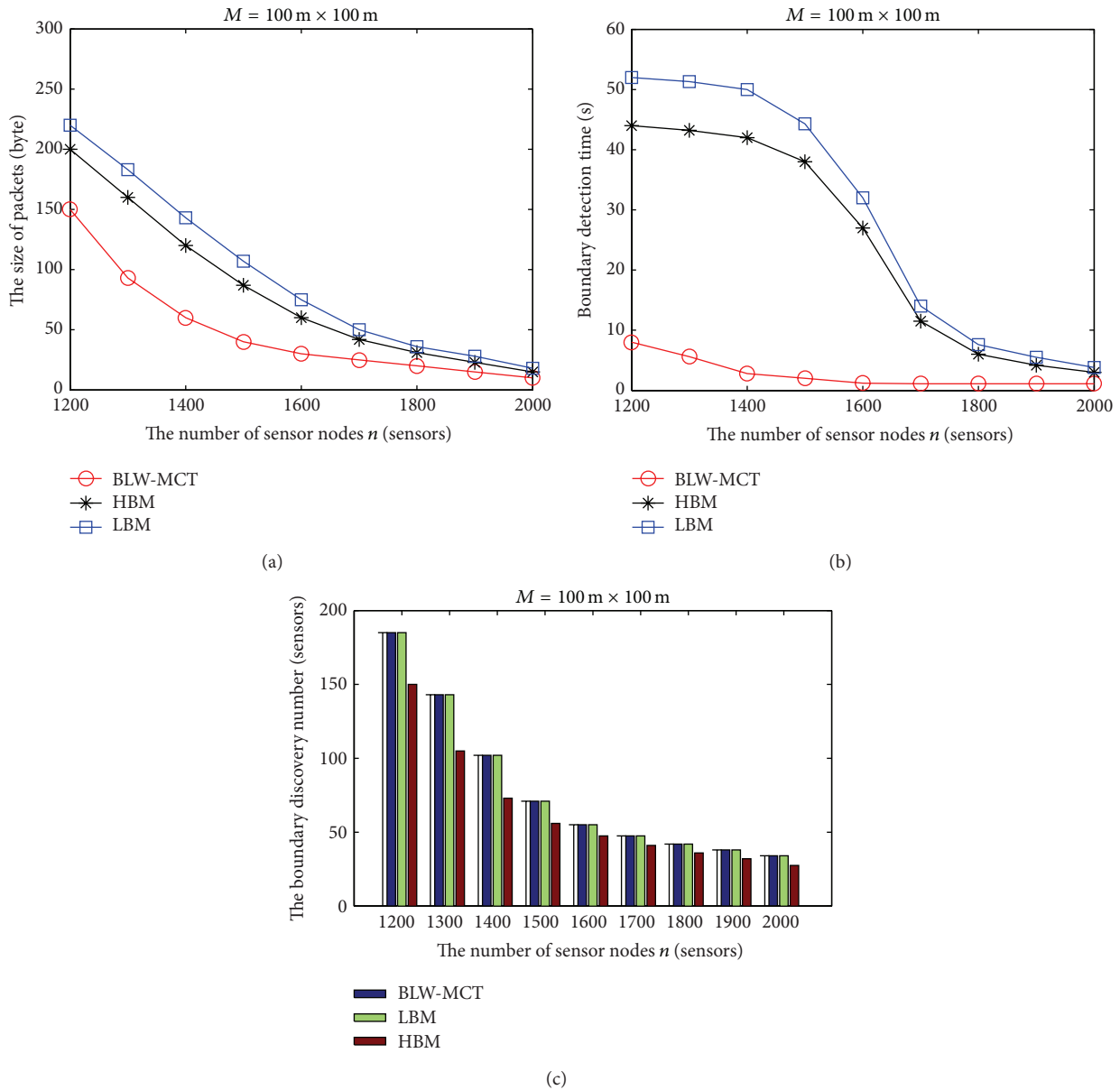


FIGURE 5: Performance comparison of different number of sensors: (a) size of packets; (b) boundary detection time; (c) boundary discovery number.

total number of coverage hole boundaries detected by the boundary detection method. It can be seen from the figure that the boundary discovery numbers of BLW-MCT, LBM, and HBM all gradually decrease with the increase of  $n$ . More specifically, the boundary discovery numbers of BLW-MCT and LBM are basically identical, which are the same as the total number of real coverage hole boundaries. But because HBM only detects the nontriangle large-scale coverage hole in WSN, the boundary discovery number of HBM is about 7% lower than the number of real coverage hole boundaries.

Figure 6 depicts the performance comparison of BLW-MCT, LBM, and HBM in terms of the size of packets, boundary detection time, and boundary discovery number with the increase of the border length of sensing region  $M$ .

During the experiment, the sensor node density in sensing region is limited to  $0.12 \text{ sensors/m}^2$ , and the sensing region varies from  $100 \text{ m} \times 100 \text{ m}$  to  $600 \text{ m} \times 600 \text{ m}$ . Since the sensor node density is limited, the number of sensor nodes will change with the increase of  $M$ . For example, when  $M$  is equal to  $100 \text{ m}$ , that is, sensing region is  $100 \text{ m} \times 100 \text{ m}$ , there are  $10000 \times 0.12 = 1200$  sensors deployed in this sensing region.

Unlike the simulation result in Figure 5, the simulation result in Figure 6 studies the network scale effects on the three evaluation parameters mentioned above. It can be seen from the figure that the three evaluation parameters mentioned above of BLW-MCT, LBM, and HBM all obviously increase with the increasing of  $M$ . In Figures 6(a) and 6(b), the size of packets and the boundary detection time all

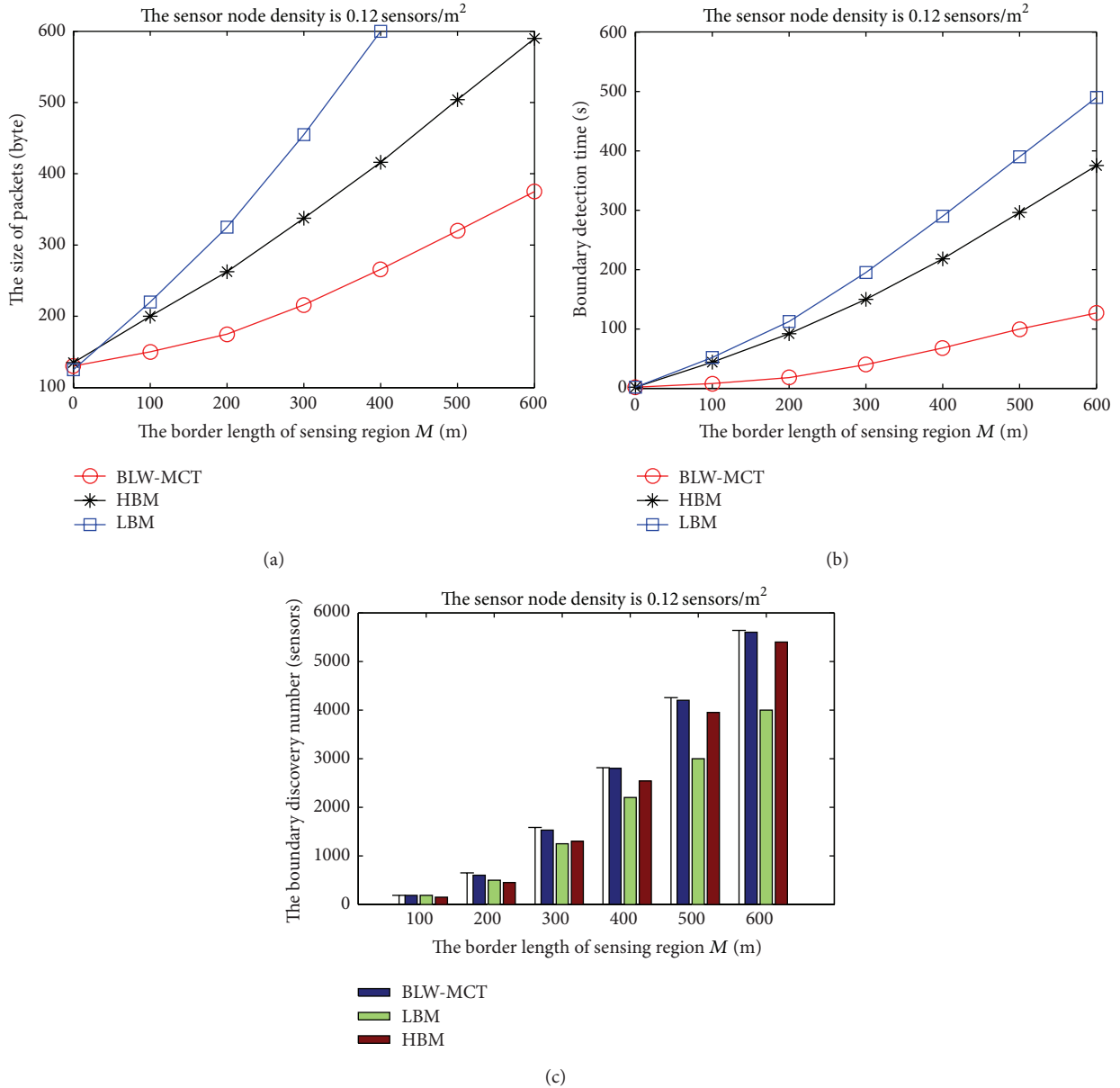


FIGURE 6: Performance comparison of different border length in sensing area: (a) size of packets; (b) boundary detection time; (c) boundary discovery number.

rapidly increase with the increasing of  $M$ . This is because sensing region grows exponentially while  $M$  grows linearly, which will inevitably lead to an increase in the number of coverage holes in WSN. On the basis of that, combining with the cause analysis for Figure 5(a), we will deduce that the size of packets and boundary detection time of BLW-MCT are less than LBM and HBM, but the size of packets and boundary detection time of HBM are slightly less than LBM. In Figure 6(c), boundary discovery numbers all obviously increase with the increasing of  $M$ . When  $M$  is less than 150 m, the total boundary discovery numbers of BLW-MCT and LBM are basically identical, which are the same as the total real coverage hole boundaries. As the  $M$  increased to around 250 m, the total boundary discovery number of BLW-MCT is the same as the total real coverage hole boundaries,

but the total boundary discovery number of LBM is less than that of BLW-MCT. When the  $M$  increases to more than 300 m, the total boundary discovery number of BLW-MCT basically remains the same as the total real coverage hole boundaries, but the total boundary discovery number of LBM may be less than that of HBM. This is because the position error of LBM will rapidly increase with the increasing of  $M$ , which directly decreases the total number of coverage hole boundaries detected by LBM.

## 6. Conclusions

The large-scale coverage hole boundary detection plays an important role in exploiting application fields for wireless

sensor network and improving the response speed for large-scale coverage hole recovery. We carry out some beneficial exploration and try in this respect and propose a boundary detection method for large-scale coverage holes in WSN based on minimum critical threshold constraint. On the basis of existing researches, we introduce the concept of minimum critical threshold, which can define large-scale coverage holes range and reduce the number of coverage holes detected by boundary detection method; we propose local-scale rough boundary detection and its iterative thinning algorithm, which can bring the minimum critical threshold organically into large-scale coverage hole boundary detection to detect the accurate large-scale coverage hole boundary in WSN quickly and efficiently. Comparing the BLW-MCT presented in this paper with LBM and HBM, the experimental results show that the computational complexity and network overhead of BLW-MCT outperform that of LBM and HBM under keeping the rates of missing coverage hole boundary almost invariant. Besides, BLW-MCT has better performance in network scalability with the increase of network scale, which will not only help to improve the efficiency of large-scale coverage hole boundary detection in large-scale WSN but will also lay a foundation for using the external mobile device to recover large-scale coverage hole.

### Conflict of Interests

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

### Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 61305113; the Natural Science Foundation of Hebei Province under Grant no. F2012203199; and the Doctor Fund of Hebei Normal University of Science and Technology under Grant no. 2011YB-006.

### References

- [1] 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.
- [2] J. Elson and D. Estrin, *Sensor Networks: A Bridge to the Physical World*, Kluwer Academic, Norwell, UK, 2004.
- [3] 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.
- [4] S. Sengupta, S. Das, M. Nasir, A. V. Vasilakos, and W. Pedrycz, "An evolutionary multiobjective sleep-scheduling scheme for differentiated coverage in wireless sensor networks," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1093–1102, 2012.
- [5] Z. Vyacheslav, E. Adil, A. Sergey, and C. Hyuseung, "Energy-efficient area coverage by sensors with adjustable ranges," *Sensors*, vol. 9, no. 4, pp. 2446–2460, 2009.
- [6] X. Wang, S. Han, and Y. Wu, "Coverage and energy consumption control in mobile heterogeneous wireless sensor networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 975–988, 2013.
- [7] A. Kroller, 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*, pp. 1000–1009, 2006.
- [8] 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.
- [9] 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.
- [10] C.-H. Wu, K.-C. Lee, and Y.-C. Chung, "A delaunay triangulation based method for wireless sensor network deployment," *Computer Communications*, vol. 30, no. 14, pp. 2744–2752, 2007.
- [11] A. Boukerche and F. Xin, "A voronoi approach for coverage protocols in wireless sensor networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference*, pp. 5190–5194, November 2007.
- [12] C. Zhang, Y. C. Zhang, and Y. G. Fang, "Localized algorithms for coverage boundary detection in wireless sensor networks," *Wireless Networks*, vol. 15, no. 1, pp. 3–20, 2009.
- [13] E. D. Zhao, J. Yao, H. Wang, and Y. T. Lv, "A coverage hole detection method and improvement scheme in WSNs," in *Proceedings of the International Conference on Electric Information and Control Engineering (ICEICE '11)*, pp. 985–988, April 2011.
- [14] 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.
- [15] F. Yan, P. Martins, and L. Decreusefond, "Connectivity-based distributed coverage hole detection in wireless sensor networks," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference*, pp. 1–6, December 2011.
- [16] Y. Bejerano, "Coverage verification without location information," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 631–643, 2012.
- [17] 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.
- [18] H. Chintakunta and H. Krim, "Distributed localization of coverage holes using topological persistence," *IEEE Transactions on Signal Processing*, vol. 62, no. 10, pp. 2531–2541, 2014.
- [19] J. Chen, J. Li, S. He, Y. Sun, and H.-H. Chen, "Energy-efficient coverage based on probabilistic sensing model in wireless sensor networks," *IEEE Communications Letters*, vol. 14, no. 9, pp. 833–835, 2010.
- [20] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, USA, 2010.
- [21] X. X. Ning, *Blocking Flow Theory and Its Application*, Science Press, Beijing, China, 2nd edition, 2009.
- [22] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific Press, Belmont, Canada, 2nd edition, 1999.

