Journal of Southeast University (English Edition) Vol. 35, No. 1, pp. 8 – 15 Mar. 2019 ISSN 1003—7985

Coordinate-free *k*-coverage hole detection algorithm in wireless sensor networks

Ma Wenyu[1] Yan Feng[1] Zuo Xuzhou[2] Xia Weiwei[1] Shen Lianfeng[1]

([1]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China)

([2]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: For wireless sensor networks, a simple and accurate coordinate-free *k*-coverage hole detection scheme is proposed. First, an algorithm is presented to detect boundary cycles of 1-coverage holes. The algorithm consists of two components, named boundary edge detection and boundary cycle detection. Then, the 1-coverage hole detection algorithm is extended to *k*-coverage hole scenarios. A coverage degree reduction scheme is proposed to find an independent covering set of nodes in the covered region of the target field and to reduce the coverage degree by one through sleeping those nodes. Repeat the 1-coverage hole detection algorithm and the higher order of coverage holes can be found. By iterating the above steps for $k-1$ times, the boundary edges and boundary cycles of all *k*-coverage holes can be discovered. Finally, the proposed algorithm is compared with a location-based coverage hole detection algorithm. Simulation results indicate that the proposed algorithm can accurately detect over 99% coverage holes.

Key words: *k*-coverage hole detection; *k*-coverage; wireless sensor networks

DOI: 10.3969/j.issn.1003 – 7985.2019.01.002

Wireless sensor networks have attracted much attention due to their wide potential applications in environmental monitoring, military defense and so on. Many of these applications require that the target field is fully covered by sensors. For the applications that need stronger environment monitoring, such as military application, triangulation-based positioning or for fault-tolerant purpose[1], it is necessary to ensure that the target field is sufficiently *k*-covered ($k \geqslant 2$), which means that every point in the area is at least covered by *k* sensors. However, coverage holes may exist due to many reasons, such as random deployment, energy depletion or destruction of sensors. It is thus essential to detect and localize all coverage holes in order to ensure the full operability of the network.

In this paper, we propose an accurate range-based *k*-coverage hole detection algorithm to discover the boundary cycles of all coverage holes, where *k* is a predefined integer and $k \geqslant 1$.

We present a scheme for discovering 1-coverage holes. The WSN is modeled as a graph and each edge of the graph is checked to determine whether it is a boundary edge. Then, we go further to find boundary cycles to localize each coverage hole by broadcasting messages along the boundary edges. After that, we extend the 1-coverage hole detection algorithm to *k*-coverage scenarios. A coverage degree reduction algorithm is proposed to find an independent covering set of nodes for the covering region of the target field, and sleep these nodes to reduce the coverage degree by one. For the remaining network, we continue to find all the boundary edges and boundary cycles. Iterating the coverage degree reduction and 1-coverage hole detection algorithm for $k-1$ times, we can find the boundary edges and boundary cycles of all *k*-coverage holes.

The complexity of the proposed algorithm is analyzed and we compare our algorithm with an idealized location-based approach under $k=1$ and $k=2$ situations. It is shown that our algorithm can accurately find boundary cycles of over 99% coverage holes under different node densities.

1 Related Work

Extensive research has been dedicated to coverage problems in WSNs. Many coverage hole detection approaches have been proposed and they can be classified into three categories: location-based, range-based and connectivity-based approaches.

Location-based schemes usually use computational geometry approaches for discovering coverage holes. In Ref. [2], a distributed coverage hole detection protocol using location information of each sensor's one and two-hop neighbors was designed. An accurate location-based method was proposed to discover the boundaries of all 1-coverage holes[3]. Various methods[4-6] for coverage hole detection based on Voronoi diagrams and Delaunay triangles were proposed. Huang et al.[1] proved that the node does not border any coverage holes if its sensing border is perimeter-covered by other sensors, and a *k*-coverage ver-

Received 2018-08-25, **Revised** 2018-11-12.

Biographies: Ma Wenyu(1994—), female, graduate; Yan Feng(corresponding author), male, doctor, associate professor, feng.yan@seu.edu.cn.

Foundation item: The National Natural Science Foundation of China (No. 61601122, 61471164, 61741102).

Citation: Ma Wenyu, Yan Feng, Zuo Xuzhou, et al. Coordinate-free *k*-coverage hole detection algorithm in wireless sensor networks[J]. Journal of Southeast University (English Edition), 2019, 35(1): 8 – 15. DOI: 10.3969/j.issn.1003 – 7985.2019.01.002.

ification scheme based on perimeter coverage was also presented. However, location-based methods require precise coordinate information of all sensors, which limits their applicability since acquiring accurate sensor location is either expensive or impractical[7]. The range-based approaches attempt to find coverage holes using localized distance information between neighboring sensors. In Refs. [8 − 9], a k-coverage verification scheme utilizing the sensing border concept was proposed for discovering boundary nodes. Qiu and Shen[10] put forward a Delaunay-based coordinate-free mechanism to detect coverage holes. In Ref. [11], the authors proposed two algorithms based on localized Voronoi polygon and neighbor embracing polygon for k-coverage holes detection, in which distance and angular information are both necessary. A k-coverage verification algorithm that used a divide-and-conquer approach based on a dimension reduction mechanism was proposed in Ref. [12]. In Ref. [13], a range-based scheme was proposed for discovering boundary edges and boundary cycles of 1-coverage holes. Connectivity-based approaches attract a great deal of attention since they only need connectivity information which is easy to obtain. A distributed coverage hole detection algorithm based on computation of a certain generator of the first homology of the Rips complex of the network was proposed[14]. Yan et al. [15] presented a method based on the homology theory for discovering all non-triangular coverage holes. However, connectivity-based approaches are not capable of finding all coverage holes.

Although there are many excellent works on coverage hole discovery, there are few schemes that can discover the boundary edges and boundary cycles of coverage holes at the same time except Refs. [3, 11, 13]. However, the methods in Refs. [3, 11] need position or angular information which is difficult to obtain. In Ref. [13], we only discovered boundary cycles of 1-coverage holes. It is thus essential to design an efficient k-coverage hole detection algorithm only using localized distance information.

2 Network Model

2.1 Models and assumptions

Consider a WSN comprised of a set of sensors (also called nodes) $V = \{v_1, v_2, \ldots, v_N\}$, and all the nodes are isomorphic. The sensors are randomly deployed in a planar target field. Each sensor is capable of monitoring a region within a circle of sensing radius R_s and communicating with adjacent nodes within its communication radius R_c. It was proved in Ref. [16] that coverage implies connectivity if and only if $R_c \geqslant 2R_s$. In the rest of the paper we set $R_c = 2R_s$ for simplicity. Our scheme is also applicable for $R_c > 2R_s$.

Nodes v_i and v_j ($v_i, v_j \in V$ and $i \neq j$) are neighbors if and only if the line segment $v_i v_j \leqslant R_c$. Each node v_i is oblivious to its location but can estimate the distance to its neighbors and share the information with them, e. g. by using methods presented in Ref. [17].

In addition, some other assumptions are as follows:

1) As shown in Fig. 1, the target field is a continuous region and the sensing range of each node v_i is denoted as a disk D_i. No two sensors are located in the same position and each node has a unique ID number.

2) Similar to Ref. [8], we assume that fence of the target field is already covered by nodes. Sensors located in the fence are periphery nodes and the others are interior nodes. Each sensor is not aware of its location but knows whether it is a periphery or an interior node by using the method proposed in Ref. [8]. In this study, we only need to verify the coverage of the interior nodes.

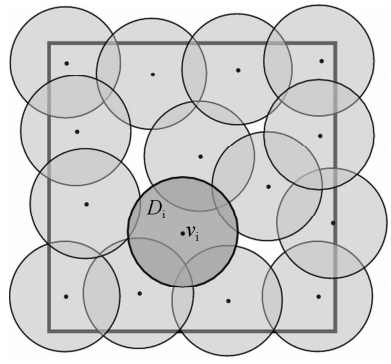


Fig. 1 Network model

2.2 Definitions

A wireless sensor network can be represented as a graph $G = (V, E)$, where V is the set of nodes in the given wireless sensor network and E can be defined as

$$E = \{v_i v_j : v_i, v_j \in V \wedge v_j \in N_i\} \tag{1}$$

where N_i is the set of nodes neighboring to node v_i.

A region Λ is fully k-covered by nodes if

$$\begin{aligned} &\forall p \in \Lambda, \ \exists \hat{V} \subseteq V \qquad |\hat{V}| = k \\ &\text{s. t.} \qquad p \in \bigcap_{v_i \in \hat{V}} D_i \end{aligned} \tag{2}$$

We define the k-coverage holes as a continuous area covered by at most $k - 1$ sensors.

Considering two neighboring nodes v_i, v_j deployed in the target field, the Euclidean distance between them is denoted by $d_{i,j}$. Sensing borders of these two neighboring nodes denoted by C_i and C_j may intersect at two points, that is

$$C_i \cap C_j = \{p_{i,j}, p'_{i,j}\} \qquad v_j \in N_i \tag{3}$$

Particularly, there is only one crossing point when $d_{i,j} = 2R_s$.

It was proved in Ref. [1] that a node is not adjacent to any coverage holes if its sensing border is perimeter-covered by other sensors. This property can be further adopted as follows.

Property 1 For any node $v_j \in N_i$, $C_i \cap C_j = \{p_{i,j},$

$p'_{i,j}\}$. Edge $v_i v_j$ is not adjacent to any coverage hole if

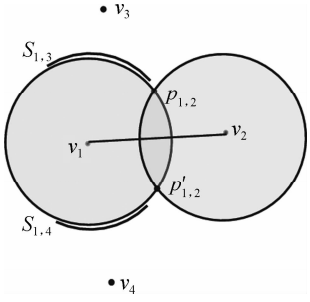
$$\exists\, v_k, v_q \in N_{i,j}$$
$$\text{s. t.} \quad p_{i,j} \in D_k \wedge p'_{i,j} \in D_q$$

where $N_{i,j}$ is the common neighbors of node v_i and v_j.

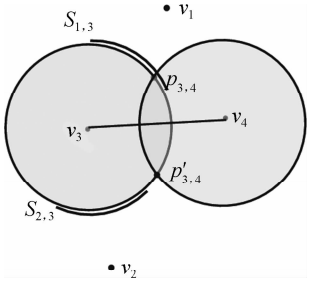
Definition 1 (boundary edge) For any edge $v_i v_j$, if the crossing points of C_i and C_j are not covered by common neighbors of them, we call edge $v_i v_j$ a boundary edge.

Definition 2 (weight of edge) For any edge $v_i v_j$, if neither of its associate crossing points is covered by other nodes, its weight w_{ij} is 2. If only one crossing point is covered by others, its weight w_{ij} is 1. If both the crossing points are covered, w_{ij} equals 0.

As shown in Fig. 2(a), edge $v_1 v_2$ is a boundary edge of weight $w_{12} = 2$ since neither of its associate crossing points is covered by other nodes. While the weight of boundary edge $v_3 v_4$ in Fig. 2(b) is 1 since only one of its associate crossing points is covered.



(a)



(b)

Fig. 2 Boundary edges of different weights. (a) $w_{12} = 2$; (b) $w_{34} = 1$

Definition 3 (redundant node) For a node v_i located in the network, if node v_i's sensing disk is fully covered by other nodes, v_i is regarded as a redundant node.

Definition 4 (boundary cycle) For each coverage hole, the boundary cycle is defined as the cycle that sequentially connects all boundary edges bordering the same coverage hole.

We can see from Fig. 3 that there are five boundary edges bordering the same coverage hole. The boundary cycle of the coverage hole is $[1, 2, 3, 4, 5, 1]$.

3 1-Coverage Hole Detection Algorithm

In this section, we present a 1-coverage hole detection algorithm based on localized distance information. The whole process can be divided into two components: boundary edge detection and boundary cycle detection.

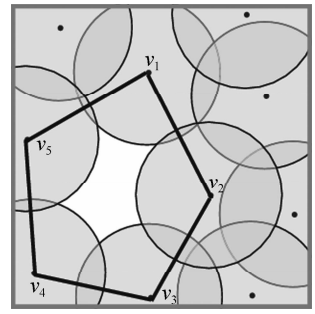


Fig. 3 Boundary cycle

3.1 Boundary edge detection

Considering two neighboring nodes v_i and v_j with distance $d_{i,j}$, as shown in Fig. 4, the segment of v_i's sensing border C_i covered by v_j's sensing disk D_j can be denoted as

$$S_{i,j} = C_i \cap D_j \qquad v_j \in N_i \tag{4}$$

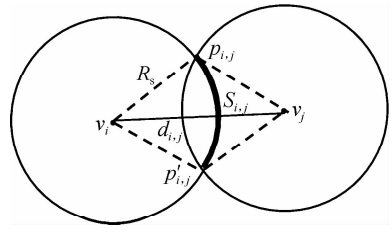


Fig. 4 Intersection segment of node v_i and v_j

The length of segment $S_{i,j}$ denoted by $\mu_{i,j}$ is

$$\mu_{i,j} = 2\cos^{-1}\frac{R_s^2 + d_{i,j}^2 - R_s^2}{2d_{i,j}R_s} = 2\cos^{-1}\frac{d_{i,j}}{2R_s} \tag{5}$$

To determine whether the crossing points are covered by common neighbors, we can check the geometrical relations between $S_{i,j}$ and other intersect segments.

Assume that segments $S_{i,k}$ is the section that node v_k's sensing border intersects with node v_i. The angle θ_{jik} between edge $v_i v_j$ and edge $v_i v_k$ is

$$\theta_{jik} = \cos^{-1}\frac{d_{i,j}^2 + d_{i,k}^2 - d_{j,k}^2}{2d_{i,j}d_{i,k}} \tag{6}$$

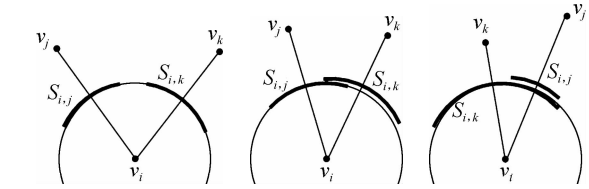
There are three different relationships between $S_{i,j}$ and $S_{i,k}$, which are useful for boundary edges detection, as illustrated in Figs. 5(a) to (c). Notice that when $S_{i,k} \subset S_{i,j}$, it is irrelevant for discovering boundary edges.

1) Disjoint: Neither of the two crossing points is covered by node v_k, as illustrated in Fig. 5(a).

2) Overlap: Only one crossing point is covered by v_k, as illustrated in Fig. 5(b).

3) Subsume: Both the two crossing points of node v_i and v_j are covered by node v_k, as illustrated in Fig. 5(c).

We can reach the following corollaries according to the geometrical relationship:



(a) (b) (b)

Fig. 5 Intersection relationships. (a) Disjoint; (b) Overlap; (c) Subsume

Corollary 1 For a node $v_k \in N_i$, if $\mu_{i,k} \geqslant \mu_{i,j} + 2\theta_{jik}$, then $p_{i,j}, p_{i,j}' \in D_k, k \neq i, j$.

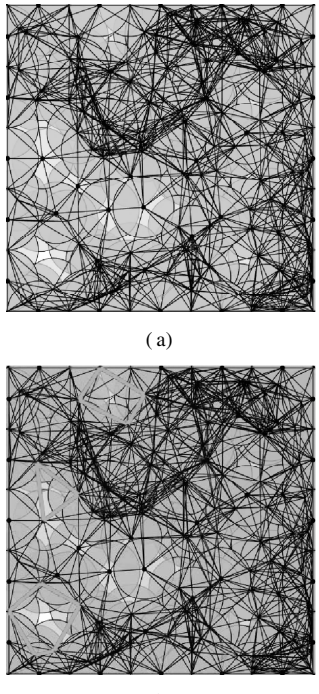
Corollary 2 For a node $v_k \in N_i$, if $\theta_{jik} > \left|\frac{\mu_{i,j} - \mu_{i,k}}{2}\right| \wedge \theta_{jik} \leqslant \frac{\mu_{i,j} + \mu_{i,k}}{2}$, then $p_{i,j} \in D_k \wedge p_{i,j}' \notin D_k$.

Corollary 3 For any two nodes v_k and v_q that cover only one of the associated crossing points of edge $v_i v_j$, they are located at different sides of $v_i v_j$ when any one of the following conditions is satisfied:

1) $v_q \notin N_k$;

2) $v_q \in N_k$ and $\theta_{kiq} = \theta_{jik} + \theta_{jiq} \vee \theta_{kiq} + \theta_{jik} + \theta_{jiq} = 2\pi$.

For each edge $v_i v_k$, check the geometrical relationships between nodes v_i, v_j and their common neighbors by Corollaries 1 to 3, and then all the boundary edges and their weights can be found. As shown in Fig. 6(a), sensors are randomly distributed in the target field and each black line connects two neighboring nodes. After the boundary edge detecting algorithm, all the boundary edges of 1-coverage holes in the network can be discovered. As shown in Fig. 6(b), the bold lines denote the discovered boundary edges.



(a)

(b)

Fig. 6 Discover boundary edges in the target field. (a) A given WSN; (b) Discovered boundary edges

Algorithm 1 Boundary edge detection

```
Begin
for each edge v_i v_j ∈ E do
  if N_{i,j} = ∅ then
    w_ij ← 2
  else
  for each node v_k ∈ N_{i,j} do
    if v_k satisfies Corollary 1 then
      flag_cover ← 1
      break
    else if v_k satisfies Corollary 2 then
      N_1 = N_1 + {v_k}
    end if
  end for
  if flag_cover = 0 then
    if |N_1| = 0 then
      w_ij ← 2
    else if |N_1| = 1 then
      w_ij ← 1
    else
      for any two nodes v_k, v_q ∈ N_1 do
        if v_k, v_q satisfies Corollary 3 then
          flag_pair ← 1
          break
        end if
      end for
      if flag_pair = 0 then
        w_ij ← 1
      end if
    end if
  end if
end if
end for
End
```

3.2 Boundary cycle detection

Once all boundary edges and their weights are found, each relevant node sends information to its corresponding neighbor. For example, node v_i discovers that edge $v_i v_j$ is a boundary edge with weight 1 or 2, it sends the message to node v_j and v_j keeps this information. However, there may be adjacent holes under lower node density. It is not enough to determine the exact boundary of each coverage hole only by the information of boundary edges. We need further knowledge about boundary cycles to localize each coverage hole. This work can be done by broadcasting messages along boundary edges.

Property 2 For any boundary edge $v_i v_j$, if $w_{ij} = 1$, $v_i v_j$ borders one coverage hole; if $w_{ij} = 2$, $v_i v_j$ borders two coverage holes.

First, we choose the initial edge and node to begin the process of broadcasting a message along boundary edges. To improve the accuracy of discovering the boundary cycle and avoid message overheard in the broadcasting stage, the

following rule for selecting the initial edge is proposed. Notice that there can be three different positional relationships between any two coverage holes. If two coverage holes have a common boundary edge, we choose the common edge as the initial edge. If two coverage holes have a common point, choose an edge that connects with the common point as the initial edge. For coverage holes that are separated, choose any boundary edge as the initial edge. When a proper initial edge v_iv_j and initial node v_i are selected, node v_j sends a message to its boundary neighbors except node v_i. The message contains the IDs of node v_i and node v_j. The boundary neighbors continue this process until the initial node v_i receives this message.

When a node v_k receives a message from its boundary neighbor, several verifications need to be done to decide how to deal with this message. If it has already received a message that came from the same initial edge, it ignores the second message for repetition. If not, node v_k adds its ID to the end of the message and broadcasts the message to its boundary neighbors. When the initial node receives a message that begins with itself, it means that the message goes through a cycle and returns to the initial node. Then it needs to verify whether a correct cycle is found. If the initial node has already found a cycle and the second cycle resembles the first one, then ignores the second one. If not, a correct cycle is found and all boundary edges that compose this cycle reduce their weights by 1. When all messages return to the initial edge, find the next initial edge and initial node for another round of discovery. If there is no boundary edge with weight larger than 0 that can be chosen as the initial edge, the algorithm terminates.

Algorithm 2 Boundary cycle detection

```
Begin
for each initial edge v_i v_j do
  V_1 ← V_1 + {v_j}
  Msg←[i, j]
  while |V_1| > 0 do
    V_2 ← V_1
    V_1 ← ∅
    for each node v_k ∈ V_2 do
      for each boundary edge v_k v_q (v_q ∈ N_k) do
        if Msg is an old message for v_k then
            continue
        else if q = i then
          a boundary cycle C is found
          for each edge v_m v_n ∈ C
            w_mn ← w_mn − 1
          end for
        else
          V_1 ← V_1 + {v_q}
          Msg←Msg + {q}
        end if
      end for
    end while
  end for
End
```

4 *k*-Coverage Hole Detection Algorithm

In this section, we extend the 1-coverage hole detection algorithm to k-coverage. First, a coverage degree reduction process is proposed to find an independent covering set of nodes in the target field, which ensure the full coverage of the area without 1-coverage holes in the network. The k-coverage hole detection algorithm is an iterating process of the 1-coverage hole detection algorithm and coverage degree reduction scheme. By iterating the above steps for $k-1$ times, boundary edges and boundary cycles of all k-coverage holes can be discovered.

First, a redundant node verification is adopted for all nodes that do not border any coverage holes. To determine whether a node v_q is redundant or not, we need to check whether the sensing disk of the node is fully covered by other sensors.

Property 3 For each $p \in D_q$, where p is the crossing point of two sensing borders C_i and C_j (v_i, $v_j \in N_q$ and $v_j \in N_i$), D_q is fully covered by other nodes if

$$\exists\, v_{k(k\neq q)} \in N_{i,j} \quad \text{s. t. } p \in D_k$$

For every two nodes' sensing borders that have a junction within v_q's sensing disk, we check if there is another node covering this junction. If every junction is also covered by at least one node other than v_q, node v_q's sensing disk is fully covered and v_q is a redundant node.

When all redundant nodes are discovered, we keep all boundary nodes and redundant nodes active. For the nodes that are neither redundant nor bordering any coverage holes, they ensure the full coverage of the area that there are no 1-coverage holes. By sleeping these nodes, we can reduce the coverage degree of the network by one. For the remaining network, repeat the boundary edge detection process. We can discover all the boundary edges of k-coverage holes by iterating $k-1$ times. For example, all the boundary edges of 1-coverage holes in the network are found as shown in Fig. 6(b), the network after coverage degree reduction is shown in Fig. 7. Then,

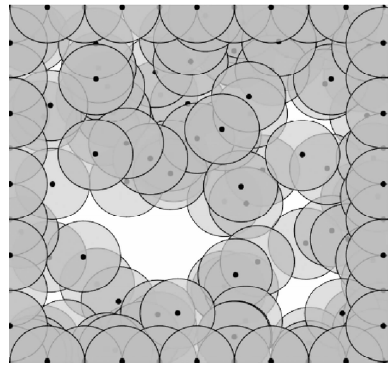


Fig. 7 Network after coverage degree reduction

all boundary edges of 2-coverage holes can be discovered by repeating the boundary edge and boundary cycle detection process in the remaining network, as shown in Fig. 8. The complete process of k-coverage hole detection is illustrated in Fig. 9.

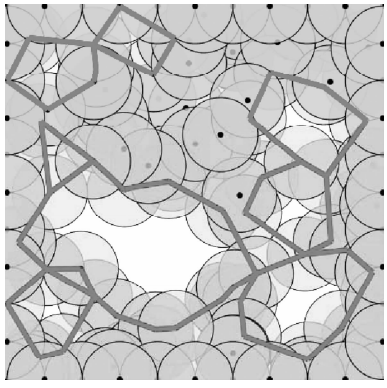


Fig. 8 2-coverage holes found in the network

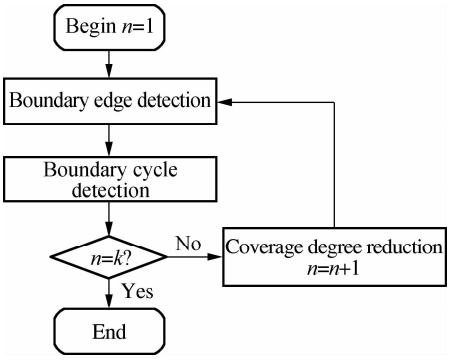


Fig. 9 k-coverage hole detection algorithm flowchart

Algorithm 3 Coverage degree reduction

```
Begin
for each active node v_q ∈ V' do
if v_q borders a coverage hole or v_q is a fence node
   V_B←v_q
else
   for any two node v_i, v_j ∈ N_q do
      if p_{i,j} ∈ D_q ∧ p'_{i,j} ∉ D_q then
         for ∀ v_m ∈ N_{i,j} ∧ m≠q do
            if p_{i,j} ∈ D_m then
               flag1←1
               break
            else
               flag1←0
            end if
         end for
      else if p_{i,j} ∈ D_q ∧ p'_{i,j} ∈ D_q then
         for ∀ v_m, v_n ∈ N_{i,j} and m, n≠q do
            if p_{i,j}, p'_{i,j} ∈ D_m or p_{i,j} ∈ D_m ∧ p'_{i,j} ∈ D_n then
               flag1←1
               break
            else
               flag1←0
            end if
         end for
      else
         continue
      end if
      if flag1 = 1
         flag_cover←1
      else
         flag_cover←0
         break
      end if
   end for
   if flag_cover = 1 then
      V_R←v_q
      sleep node v_q
   end if
end if
end for
for each v_i ∈ V' − V_B − V_R do
 sleep node v_i
end for
for each v_j ∈ V_R do
   active node v_j
end for
End
```

5 Simulation and Performance Evaluation

5.1 Complexity analysis

In the boundary edge detection part, each edge needs to check whether it is a boundary edge and compute its weight. This can be done by firstly checking the geometrical relationships between the two nodes that form the edge and their common neighbors. For each edge, the worst-case computation complexity is $O(n)$, where n is the number of neighboring nodes. Then, check the set of nodes that only cover one crossing point of the sensing borders and find whether there is a pair of nodes located at different sides of the edge. Each edge needs to check $n(n-1)$ times at most since there are maximum n nodes matching the above condition. For each node, the maximum number of connected edges is n. Thus, the computation complexity of the boundary edge detection part is $O(n^3)$.

In the boundary cycle detection part, each node just needs to broadcast the message to its boundary neighbors, and the computation complexity of this step is $O(1)$.

In the coverage degree reduction part, for each node w, every two neighbors that have a junction of their sensing borders within w's sensing range need to be checked, the worst-case computation complexity is $O(n)$. Then for each junction, we need to determine whether it is also covered by another node. The complexity of this step is $O(n^2)$. Thus, the complexity of coverage degree reduction part is $O(n^3)$. Iterate the above steps for $k-1$ times and all the boundary edges of k-coverage holes can be dis-

covered.

Therefore, the total worst-case computation complexity of the 1-coverage hole detection algorithm is $O(n^3)$. The total worst-case computation complexity to discover all k-coverage holes is $O(kn^3)$, where n is the number of neighboring nodes of each sensor.

5.2 Performance evaluation

The algorithm is realized with Matlab. A 100 m × 100 m square area is chosen as the target field. The sensing radius R_s is set to be 10 m and communication radius R_c is set to be 20 m. We assume that sensors are randomly distributed in the target field according to a Poisson point process with intensity λ, and the proposed algorithm also works for other random distributions. In order to evaluate the performance of our proposed algorithm (We denoted it as RBA), we compare it to the location-based algorithm (LBA) proposed in Ref.[3] since it has been proved that LBA can detect all the boundary cycles of coverage holes correctly. The LBA scheme needs precise coordinate information of all nodes in the network. However, as discussed in Section 1, acquiring precise location information of all sensors is impractical, so LBA is considered as an idealized algorithm.

Detection results of our algorithm (RBA) and LBA are compared under $k=1$ and $k=2$ situations. For the situation where $k>2$, we only need to increase the number of iterations. If the results of these two algorithms are identical, it is regarded as a correct decision. It is possible that there is a shorter path between two nodes in the cycle. If the results are different but can be equivalent after shortening the cycle by one hop, we still considered it as a correct decision. Otherwise, we regard it as an error.

Tab. 1 shows the total numbers of coverage holes discovered by LBA and RBA under $k=1$ and $k=2$ with different node densities. For each λ, 1 000 times simulation are performed. From the table we can see that our algorithm can accurately detect boundary cycles of over 99% coverage holes. In addition, we show the simulation results in Fig. 10 for better understanding.

Tab. 1 Numbers of coverage holes found by LBA and RBA

λ	$k=1$			$k=2$		
	LBA	RBA	Accuracy/%	LBA	RBA	Accuracy/%
0.009	11 777	11 748	99.75	13 469	13 359	99.18
0.010	10 880	10 864	99.85	13 473	13 405	99.49
0.011	9 755	9 753	99.98	12 999	12 913	99.34
0.012	8 732	8 724	99.91	12 310	12 260	99.59
0.013	7 622	7 618	99.95	11 647	11 610	99.68

6 Conclusion

In this paper, we propose an efficient coverage hole detection algorithm using only the localized distance information of neighboring nodes for wireless sensor networks. The algorithm consists of three components. In the first two components, we use a graph to model the WSN and discover boundary edges and boundary cycles of all 1-coverage holes. In the third component, we find an independent covering set of nodes in the covered region of the target field, and reduce the coverage degree of the target field by sleeping those nodes. By iterating the above steps for $k-1$ times, we can discover boundary edges and boundary cycles of all k-coverage holes. By comparing with an idealized location-based algorithm under $k=1$ and $k=2$, it is shown that our proposed algorithm can accurately discover the boundary cycles of over 99% coverage holes under different node densities. The complexity of our algorithm to discover all the k-coverage holes is $O(kn^3)$, where n is the number of neighbors of each node.

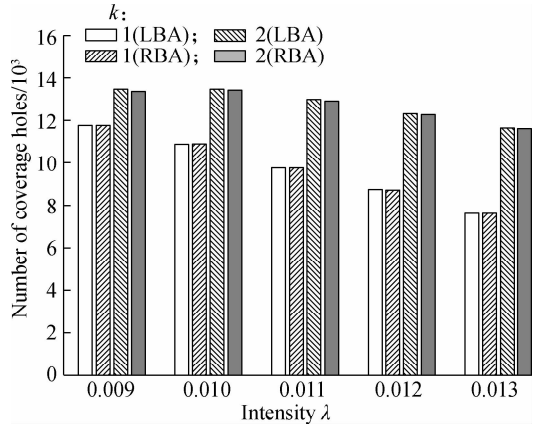


Fig. 10 Performance evaluation under $k=1$ and $k=2$

References

[1] Huang C, Tseng Y. The coverage problem in a wireless sensor network[J]. *Mobile Networks and Applications*, 2005, **10**(4): 519-528.

[2] Ma H C, Kumar Sahoo P, Chen Y W. Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks[J]. *Journal of Network and Computer Applications*, 2011, **34**(5): 1743-1756. DOI: 10.1016/j.jnca.2011.06.007.

[3] Tong B, Tavanapong W. On discovering sensing coverage holes in large-scale sensor networks, TR 06-03[R]. Ames, USA: Computer Science, Iowa State University, 2006.

[4] Qiu C, Shen H, Chen K. An energy-efficient and distributed cooperation mechanism for k-coverage hole detection and healing in WSNs[C]//2015 *IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. Dallas, USA, 2015: 73-81. DOI: 10.1109/mass.2015.115.

[5] Dai G, Chen L, Zhou B, et al. Coverage hole detection algorithm based on Voronoi diagram in wireless sensor network[J]. *Journal of Computer Applications*, 2015, **35**(3): 620-623.

[6] Li W, Zhang W. Coverage hole and boundary nodes detection in wireless sensor networks[J]. *Journal of Network and Computer Applications*, 2015, **48**: 35-43. DOI: 10.1016/j.jnca.2014.10.011.

[7] Niculescu D. Positioning in ad hoc sensor networks[J]. *IEEE Network*, 2004, **18**(4): 24 - 29. DOI: 10.1109/mnet.2004.1316758.

[8] Bejerano Y. Simple and efficient *k*-coverage verification without location information [C]//2008 *IEEE INFOCOM*. Phoenix, USA, 2008: 897 - 905. DOI: 10.1109/infocom.2008.67.

[9] Bejerano Y. Coverage verification without location information[J]. *IEEE Transactions on Mobile Computing*, 2012, **11**(4): 631 - 643. DOI: 10.1109/tmc.2011.85.

[10] Qiu C X, Shen H Y. A delaunay-based coordinate-free mechanism for full coverage in wireless sensor networks [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2014, **25**(4): 828 - 839. DOI: 10.1109/tpds.2013.134.

[11] Zhang C, Zhang Y C, Fang Y G. Localized algorithms for coverage boundary detection in wireless sensor networks[J]. *Wireless Networks*, 2009, **15**(1): 3 - 20. DOI: 10.1007/s11276-007-0021-1.

[12] Kasbekar G S, Bejerano Y, Sarkar S. Generic coverage verification without location information using dimension reduction[J]. *ACM Transactions on Networking*, 2012, **20**(6): 1991 - 2004. DOI: 10.1109/tnet.2012.2190620.

[13] Ma W, Yan F, Zuo X, et al. Coverage hole detection algorithm without location information in wireless sensor networks[C]//2017 *International Conference on Computer and Communications*. Chengdu, China, 2017: 357 - 361.

[14] Tahbaz-Salehi A, Jadbabaie A. Distributed coverage verification in sensor networks without location information [J]. *IEEE Transactions on Automatic Control*, 2010, **55**(8): 1837 - 1849. DOI: 10.1109/tac.2010.2047541.

[15] Yan F, Vergne A, Martins P, et al. Homology-based distributed coverage hole detection in wireless sensor networks[J]. *ACM Transactions on Networking*, 2015, **23**(6): 1705 - 1718. DOI: 10.1109/tnet.2014.2338355.

[16] Wang X, Xing G, Zhang Y, et al. Integrated coverage and connectivity configuration in wireless sensor networks [C]//2013 *International Conference on Embedded Network Sensor Systems*. New York, USA, 2003: 28 - 39.

[17] Mistry H P, Mistry N H. RSSI based localization scheme in wireless sensornetworks: A survey[C]//2015 *International Conference on Advanced Computing and Communication Technologies*. Haryana, India, 2015: 647 - 652.

无线传感器网络中无坐标信息的 *k*-覆盖空洞检测算法

马文钰[1] 燕 锋[1] 左旭舟[2] 夏玮玮[1] 沈连丰[1]

([1]东南大学移动通信国家重点实验室,南京 210096)

([2]电子科技大学信息与软件工程学院,成都 610054)

摘要:针对无线传感器网络,提出了一种简单精确且无需坐标信息的 *k*-覆盖空洞检测算法. 首先,提出一种 1-覆盖空洞检测算法,算法由边界线段检测和边界圆周检测 2 部分组成. 然后,扩展算法至 *k*-覆盖空洞场景. 通过在已被节点覆盖的目标区域内寻找一独立覆盖的节点子集,并休眠该集合内的节点,使得网络覆盖度减 1. 此后,重复 1-覆盖空洞检测算法,发现更高阶的覆盖空洞. 迭代上述步骤 $k-1$ 次,可以发现所有 *k*-覆盖空洞的边界线段和边界圆周. 最后,将所提算法与基于坐标的覆盖空洞检测算法进行对比,仿真结果显示,所提算法可以精确检测 99% 以上的覆盖空洞.

关键词:*k*-覆盖空洞检测;*k*-覆盖;无线传感器网络

中图分类号:TN915