Hindawi
Journal of Sensors
Volume 2020, Article ID 3236970, 13 pages
https://doi.org/10.1155/2020/3236970

Research Article

Efficient Coverage Hole Detection Algorithm Based on the Simplified Rips Complex in Wireless Sensor Networks

Jing Zhang, Han Chu, and Xin Feng

College of Computer Science And Technology, Changchun University of Science and Technology, Changchun 130022, China

Correspondence should be addressed to Xin Feng; fengxin@cust.edu.cn

Received 21 September 2019; Accepted 7 January 2020; Published 24 February 2020

Academic Editor: Antonio Lazaro

The appearance of coverage holes in the network leads to transmission links being disconnected, thereby resulting in decreasing the accuracy of data. Timely detection of the coverage holes can effectively improve the quality of network service. Compared with other coverage hole detection algorithms, the algorithms based on the Rips complex have advantages of high detection accuracy without node location information, but with high complexity. This paper proposes an efficient coverage hole detection algorithm based on the simplified Rips complex to solve the problem of high complexity. First, Turan's theorem is combined with the concept of the degree and clustering coefficient in a complex network to classify the nodes; furthermore, redundant node determination rules are designed to sleep redundant nodes. Second, according to the concept of the complete graph, redundant edge deletion rules are designed to delete redundant edges. On the basis of the above two steps, the Rips complex is simplified efficiently. Finally, from the perspective of the loop, boundary loop filtering and reduction rules are designed to achieve coverage hole detection in wireless sensor networks. Compared with the HBA and tree-based coverage hole detection algorithm, simulation results show that the proposed hole detection algorithm has lower complexity and higher accuracy and the detection accuracy of the hole area is up to 99.03%.

1. Introduction

The Internet of things is deeply applied to social life in the form of smart cities and Internet of Vehicles. As an underlying technology of the Internet of things, wireless sensor networks (WSNs) consist of numerous sensor nodes deployed in the monitoring area for comprehensively sensing, acquiring, and transmitting information about objects, which is suitable for intelligent transportation [1], event detection [2], environmental monitoring [3], etc. These practical applications have high requirements for the service quality of WSNs. The coverage rate is an important metric for evaluating WSNs' service quality [4], and many scholars have contributed to its improvement and optimization. In [5], an insightful and comprehensive summarization and classification on the data fusion-based coverage optimization problem and techniques are provided aiming at overcoming the shortcomings existing in current solutions. The scaling laws between coverage, network density, and SNR are derived in [6], and data fusion is shown to significantly improve sensing coverage by exploiting the collaboration among sensors. However, some reasons lead to the loss of coverage in the network inevitably, such as random deployment, location modification, and energy exhaustion of the nodes. Thus, data lost or information undelivered will occur in the uncovered area of the original network, which degrades the service quality of WSNs. The uncovered areas are called coverage holes [7]. The appearance of holes not only destroys the communication link and reduces the data accuracy [8] but also aggravates the transmission burden of the boundary nodes near the holes, resulting in the expansion of the hole range [9]. Therefore, discovering and locating coverage holes in the network are crucial to ensure the quality of network services.

The existing hole detection algorithms can be roughly divided into the following three categories: geometric methods, probabilistic methods, and topological methods. The geometry method uses the location information of nodes and the corresponding geometry tools such as the Voronoi diagram and the Delaunay triangulations to detect holes [10–16]. Although this method can identify the coverage holes accurately, obtaining accurate location information of the sensor nodes is very expensive and difficult. Therefore, this method is not practical. The location information is unimportant in the probabilistic method [17–19]; however, uniform distribution of nodes or the high node density is an essential condition. Meanwhile, it is difficult to detect the coverage holes accurately. The topological method [20–22] often uses the connectivity information between nodes to detect holes without the location information of nodes and guarantees the detection accuracy. However, the complexity of these algorithms is always high and the efficiency is low. Thus, this paper proposes an efficient coverage hole detection algorithm based on the simplified Rips complex which belongs to the topology method. The proposed algorithm reduces the complexity of the algorithm and guarantees the detection accuracy. The main contributions of this paper are as follows:

(1) Redundant node sleeping. Combining Turan's theorem and the concept of the degree and clustering coefficient in a complex network, we divide the internal nodes in the network into two categories, namely, deterministic nodes and nondeterministic nodes; a redundant node determination rule is designed for sleeping redundant nodes in a distributed manner

(2) Redundant edge deletion. Combining the concept of the complete graph, we propose a method for identifying redundant edges, which can simplify the edge set in the network. By means of the simplification of nodes and edges, the network model that is the Rips complex can be simplified efficiently and quickly

(3) Hole detection and boundary loop identification. Based on the simplified network structure, a method for detecting holes from the perspective of the loop is proposed. Simultaneously, the method of boundary edge identification, definition of false boundary edges, boundary loop identification, and reduction rules are given in turn

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 introduces the system model and related concept definitions. The simplified process of the Rips complex will be executed in Section 4, including the identification of redundant nodes and redundant edges. Section 5 identifies the holes. Section 6 evaluates the accuracy of the algorithm through some simulation experiments. Finally, Section 7 concludes the paper.

2. Related Works

The research on coverage holes is divided into two parts, coverage hole detection and coverage hole repair, respectively. This paper mainly researches on the coverage hole detection algorithm. The existing coverage hole detection algorithms can be roughly divided into three categories: geometric methods, probabilistic methods, and topological methods.

2.1. Geometric Methods. The geometric methods use the location information of the nodes or the relative distance between nodes and combine the corresponding geometric tools (such as Voronoi diagram and Delaunay triangulations) to identify the holes. In [12], the concept of the tree is introduced to locate and describe coverage holes; thus, the location and shape of the corresponding hole can be determined, as well as the size of the hole. However, the relative location information of the neighbor nodes must be known. In [13], an algorithm based on the Delaunay triangulations is proposed, which is combined with virtual edge-based methods to help detect coverage holes in wireless sensor networks. Compared with the existing tree-based method, this algorithm helps detect the exact size of the coverage hole with the coordinate information of the nodes, which generates extra cost. In [23], each node is recognized whether it is on a hole boundary on the basis of a local Voronoi diagram; however, mobile sensors must be employed to construct a hybrid sensor network with static sensors. In [24], the Voronoi diagram-based screening strategy is proposed to screen out the boundary nodes and the exact location of the coverage holes is obtained according to the virtual edge-based hole location strategy. High detection accuracy is guaranteed by considering irregularities of the shape of the coverage holes, and more accurate information is provided for repairing the coverage holes, but the specific location information of each node must be known in this method. In [25], a method based on Delaunay is proposed to detect coverage holes without the nodes' coordinate information; however, a global view of holes cannot be given.

2.2. Probabilistic Methods. With uniform distribution of nodes and high node density, coverage holes can be detected in the network from statistical attributes. An algorithm is presented for determining the boundary node structure of a region in [26], but high-density nodes are required. Assume that the connectivity between nodes is determined by the unit disk graph model in [27], and a linear-time algorithm is proposed to identify the boundary of the holes. But the algorithm cannot distinguish two holes that are close to each other. In [28], the coverage of mobile and heterogeneous wireless sensor networks is studied, and the coverage problem under the Poisson deployment scheme with a 2-D random walk mobility model is discussed. The coverage rate will be improved by bringing in mobility. However, the coverage range is ignored.

2.3. Topological Methods. Topological methods use topological attributes (such as connectivity information) to identify the boundaries of a hole without the exact location of nodes. In [29], the combinatorial Laplacians are the right tools to compute distributed homologous groups. Although distributed hole detection can be performed, the holes cannot be located accurately. In [20], a distributed algorithm is proposed by using the simplicial complex and the combinatorial Laplacians to obtain the topological properties of the network, which verifies the existence of coverage holes in the sensor network without any metric information. But the hole boundary cannot be found accurately, because the Rips complex cannot always detect all coverage holes. In [21], the holes are first defined as triangular and nontriangular holes to study the accuracy of using the Rips complex when detecting holes, and a connectivity-based distributed hole detection algorithm is proposed for nontriangular holes, which is more suitable for nondense sensor networks. On the basis of the literature [21], the percentage of a triangular hole area under different ratios of communication radius and sensing radius is researched in [22], and the conditions for accurately detecting holes with the Rips complex are given. Simultaneously, a homology-based distributed coverage hole detection method is proposed for nontriangular holes, which cannot detect all the holes, and the complexity is high. The more planar the Rips complex is, the more favorable it will be for the detection of holes; therefore, the rapid and efficient simplification of the Rips complex can effectively reduce the complexity of the algorithm.

Most of the above coverage hole detection methods utilize the binary sensor model; meanwhile, some other models are also proposed to reflect the sensing capability and detect the coverage holes. In [30], a new confident information coverage (CIC) model is proposed for field reconstruction, whose objective is to obtain reconstruction maps of some physical phenomena's attribute with a given reconstruction quality for the whole sensor field, including points that were sampled and not sampled. In [31], LCHD and LCHDRL schemes are proposed to address and study the localized confident information coverage hole detection issue (LCICHD) with the goal of finding the locations and number of the emerged coverage holes in IoT on the basis of the CIC model. Two effective heuristic CIC hole detection algorithms including the CHD without considering the nodes' residual energy and the other CHDRE taking the nodes' residual energy into account are proposed in [32] to address and study the confident information coverage hole detection problem (CICHD) on the basis of the CIC model. An energy-efficient CIC hole detection scheme, EECICHD, which fully exploits the inner spatial correlation of the radionuclides and sensors' cooperative sensing ability for improving the CIC hole detection efficiency, is proposed in [33]. However, the abovementioned algorithms always work in a centralized manner, which are not suitable for large-scale monitoring fields. Meanwhile, these algorithms partition a continuous sensing area into a series of reconstruction grids to check for the existence of holes. The boundaries of the detected holes are determined by image processing in those algorithms.

According to the above analysis, this paper proposes an efficient coverage hole detection algorithm based on the simplified Rips complex, which designs the node and edge deletion rules firstly to simplify the Rips complex efficiently and makes the Rips complex closer to planarization, and then identifies the holes in the network from a loop perspective on the basis of the simplified Rips complex structure. The proposed algorithm in this paper detects holes in a continuous sensing area and gets the boundaries of the network accurately without image processing; thus, a binary sensor model is adopted in the network for each node.

3. System Model and Related Concept Definitions

3.1. System Model. N sensor nodes are deployed in a 2-D plane. The nodes located inside the target area are internal nodes which are randomly distributed, and the other nodes which are evenly distributed on the outer boundary of the target area to ensure the full coverage are border nodes. Each node does not know the specific location information, and a node can be determined as an internal node or is not based on an initial setting. Some other conditions are set as follows:

(1) The nodes are isomorphic, the communication range (R_c) and the sensing range (R_s) of each node are equal, and $R_c = 2R_s$

(2) A binary sensor model is adopted in the network for each node

(3) Each node has a unique ID

(4) The network is connected, as shown in Figure 1

3.2. Related Concept Definitions

3.2.1. Definitions Related to Homology Theory

Definition 1 (simplex). Given a vertex set V and a positive integer k, a k-simplex S is a random subset of $k+1$ points $[v_0, v_1, v_2, \cdots, v_k]$, where $v_i \in V$ and $v_i \neq v_j$; k is set as the dimension of a simplex [22]. As shown in Figure 2, a 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron containing the interior. All $k-1$ simplices consisting of $k+1$ vertices form the surface of the k-simplex.

Definition 2 (simplicial complex). A series of simplices are parts of a simplicial complex and form the surface of the simplicial complex. The simplices must satisfy two conditions as follows: (1) each surface of the simplex in the simplicial complex must be a surface of the simplicial complex and (2) the intersection of any two simplices $s1$ and $s2$ is both the surfaces of $s1$ and $s2$. The dimension of a simplicial complex is defined as the largest dimension of any simplex contained in the simplicial complex.

Definition 3 (Rips complex). Given a finite set of points V and a fixed radius ε in R^n, the Rips complex ($R_\varepsilon(V)$) of V is an abstract simplicial complex whose k-simplices are composed of $k+1$ points in V. The distance between any two nodes in V is less than the fixed radius ε.

Suppose that $P = \{p_i\}$ is a set of sensor node locations and $S = \{s_i\}$ is a set of sensor node sensing ranges, where p_i represents the location of the i-th node and $s_i = \{x \in R^2, \|x - p_i\| \leq R_s\}$. As shown in Figure 3, the Rips complex can be formed on the basis of the above definition when the vertex set V contains six vertices.

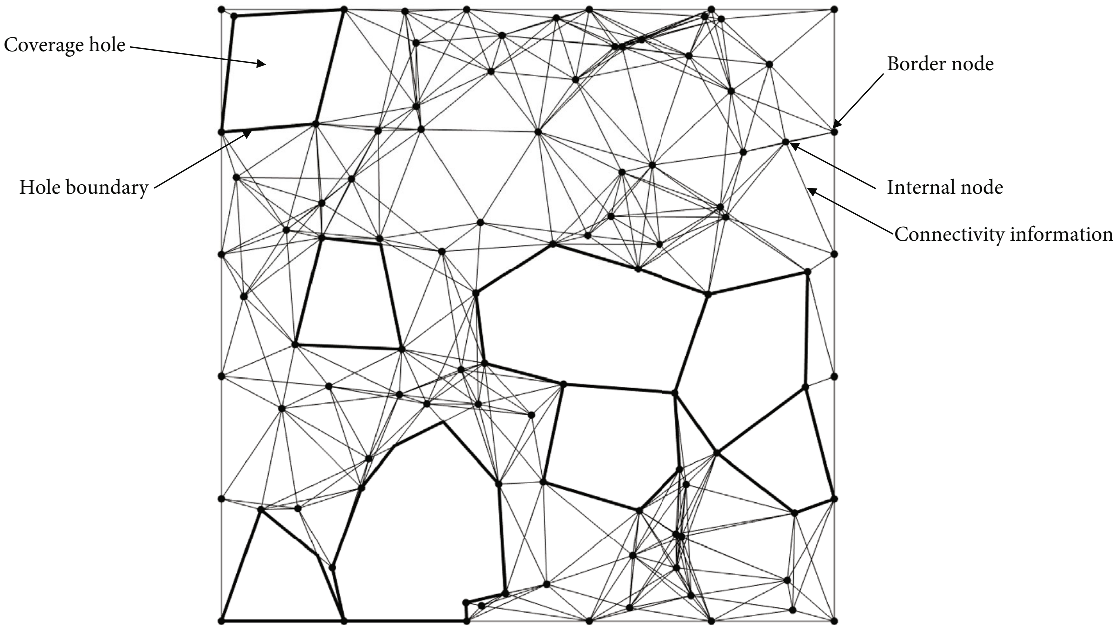


Figure 1: Network connectivity diagram.

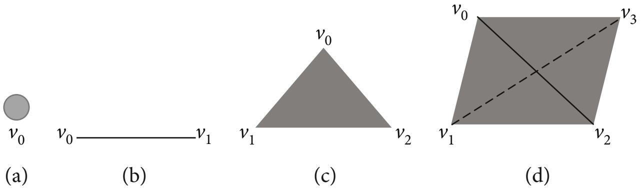


Figure 2: k-simplex: (a) 0-simplex: v_0 is a point; (b) 1-simplex: v_0 and v_1 are points; (c) 2-simplex: v_0, v_1, and v_2 are points; (d) 3-simplex: v_0, v_1, v_2, and v_3 are points.

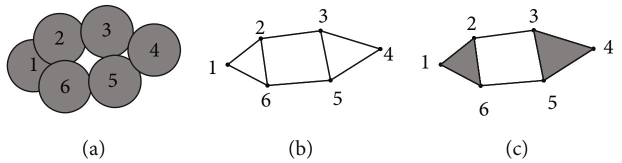


Figure 3: Example graphs: (a) coverage graph; (b) connected graph; (c) Rips complex ($R_c = 2R_s$).

Definition 4 (triangular holes and nontriangular holes [21]). The hole existing in the triangle which is not completely covered by the sensor nodes is called the triangular hole; the other holes are called nontriangular holes.

As shown in Figure 3, there are two coverage holes in the network formed by nodes 2, 3, 5, and 6 and nodes 3, 4, and 5, respectively. However, only the hole formed by nodes 2, 3, 5, and 6 can be detected by the Rips complex.

3.2.2. Definitions Related to Graph Theory. An undirected graph composed of a vertex set $V = \{v_1, v_2, \cdots, v_n\}$ and a undirected edge set $E = \{e_1, e_2, \cdots, e_m\}$ is denoted as $G = (V, E)$. Some definitions are given on the basis of the undirected graph as follows.

Definition 5 (adjacency matrix). Two nodes have an adjacent relationship if an edge between the two nodes exists; otherwise, they are not adjacent. For the convenience of calculation, an adjacency matrix A can be used to describe the relationship between the two nodes for the undirected graph $G = (V, E)$. Supposing that there are n vertexes in graph G, then $A_G = (am_{ij})_{n*n}$.

Definition 6 (subgraph). Graph G' can be expressed as (V', E'). G' is a subgraph of G, and G is called the parent graph of G' if $V' \subseteq V$ and $E' \subseteq E$ and called $G' \subseteq G$.

3.2.3. Definitions Related to Complex Network Theory. The parameters that the degree and clustering coefficient are involved in this study are defined as follows.

Definition 7 (degree). The degree k_i of the node v_i is the number of its 1-hop neighbor nodes and determined by its adjacency matrix, which can be defined as

$$k_i = \sum_{j \in N} am_{ij}, \tag{1}$$

where am_{ij} is equal to 1 when node i and node j are directly connected; otherwise, am_{ij} is equal to 0. N is the total number of nodes in the network.

Definition 8 (clustering coefficient). The clustering coefficient C_i of the node v_i is the ratio between the number of edges E_i and the total number of possible edges between the directly

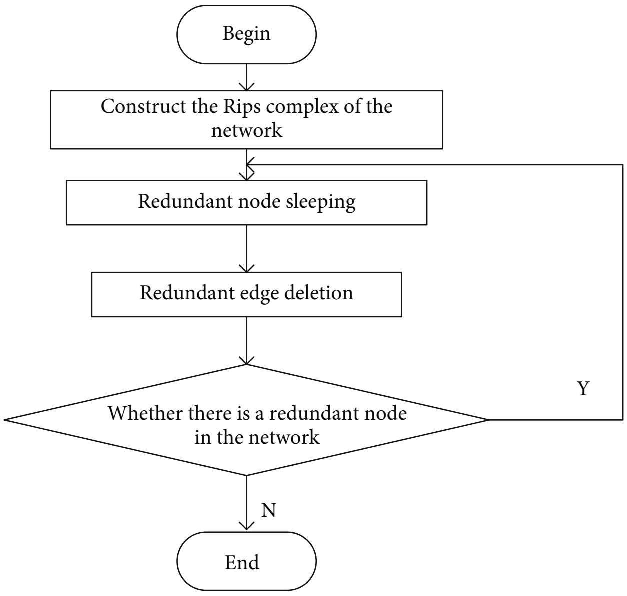

Begin
Construct the Rips complex of the network
Redundant node sleeping
Redundant edge deletion
Whether there is a redundant node in the network
Y
N
End

FIGURE 4: The flow chart of Rips complex simplification.

connected k_i neighbor nodes, and it characterizes the tightness and aggregation of nodes in the network. C_i can be defined as

$$C_i = \frac{E_i}{C_{k_i}^2} = \frac{2E_i}{k_i \cdot (k_i - 1)}, \tag{2}$$

where $C_{k_i}^2 = k_i \cdot (k_i - 1)/2$.

The node degree reflects the ability of a node to establish a direct connection with the surrounding nodes, that is, the number of the neighbor nodes, whereas the clustering coefficient presents the edges connected among neighbor nodes, that is, reflecting the tightness between the nodes.

4. Rips Complex Simplification

An efficient distributed hole detection algorithm for non-triangular holes is designed in this section. As for triangle holes, the area ratio of triangle holes in the network is less than 0.06% when the ratio of the node communication radius to the coverage radius is between $\sqrt{3}$ and 2 (including $\sqrt{3}$ and 2); that is, the triangular holes in the network can be ignored [14]. The hole detection will be more efficient and easier when the Rips complex tends to planarity; therefore, it is critical to simplify the Rips complex efficiently for reducing the complexity of the algorithm. In this paper, the redundant node determination rules and redundant edge deletion rules are constructed to make the Rips complex more planar, and then, the holes can be detected effectively. The proposed algorithm includes the following three parts: (1) redundant node sleeping, (2) redundant edge deletion, and (3) hole detection and boundary identification. The first two steps are viewed as the simplified process of the Rips complex, and the process is shown in Figure 4.

The i-simplex $[v_0, v_1, v_2, \cdots, v_i]$ is a part of j-simplex $[v_0, v_1, v_2, \cdots, v_j]$ if $[v_0, v_1, v_2, \cdots, v_i] \in [v_0, v_1, v_2, \cdots, v_j]$. Therefore, nodes v_i and v_j are part of the edge $[v_i, v_j]$, edge $[v_i, v_j]$ is part of the triangle $[v_i, v_j, v_k]$, and v_k is the neighbor node of the edge $[v_i, v_j]$. $E(v)$ is a set of edges which contains node v, and $T(v)$ is a set of 2-simplices which contains node v.

Definition 9 (loop). A loop C is a subgraph of graph G if each node on the loop C has only two neighbors. The length of loop C is the number of its edges, denoted by $|E(C)|$. All of the loops in graph G are denoted by $C(G)$, and the set of triangle loops in graph G is denoted by $C_T(G)$. The length of the triangle loop is three.

Definition 10 (neighbor graph). The neighbor set of a node v in graph G is denoted by $N_G(v)$, and the neighbor graph Γ_G (v) of node v in graph G is denoted by $G[N_G(v)]$. The node set of the neighbor graph is composed of neighbor nodes of the node v.

Definition 11 (Turan's theorem). Graph G has at most $E(r)$ edges if graph G is a simple graph with n nodes, and there is no K_{r+1} complete graph in graph G as a subgraph, where K_{r+1} represents the $(r+1)$-complete graph:

$$E(r) = \left(1 - \frac{1}{r}\right) \cdot \frac{n^2}{2}, \tag{3}$$

when $r = 2$ and $E(2) = n^2/4$.

Inference 12. There must be a K_{r+1} complete graph in graph G as a subgraph if a simple graph G with n nodes contains at least $E(r) + 1$ edges. That is, there must be a K_3 complete graph in graph G as a subgraph when $r = 2$ and graph G contains at least $E(2) + 1$ edges.

Inference 13. If each internal node in the network meets the following conditions:

$$\begin{aligned} &k_i \geq 3, \\ &C_i \geq \frac{\left(\lfloor n^2/4 \rfloor + 1\right)}{C_{k_i}^2}. \end{aligned} \tag{4}$$

Then, there must be a triangle loop in the neighbor graph of this node; otherwise, it is uncertain whether a triangle loop is in the neighbor graph of this node. k_i is the degree of the node i, C_i is the clustering coefficient of the node i, and $n = k_i$.

Proof. According to Definition 8, $E_i \geq \lfloor n^2/4 \rfloor + 1$ if the degree of a node v is greater or equal to 3 (that is, there is at least three nodes in the neighbor graph of the node v), and the clustering coefficient of the node v meets $C_i \geq (\lfloor n^2/4 \rfloor + 1)/C_{k_i}^2$. Therefore, there must be K_3 in the neighbor graph of the node v as a subgraph on the basis of Inference 12.

Definition 14 (determined and nondetermined nodes). The internal node is a determined node if there is a triangle loop in its neighbor graph; otherwise, the internal node is a nondetermined node.

Definition 15 (redundant node determination rule). Each internal node v in the network is determined as a redundant node if its neighbor graph satisfies the following two conditions: (1) the neighbor graph of node v is connected and (2) all loops can be triangulated (that is, the length of each loop is three).

Proof. In order to prove the correctness of the redundant node determination rule, it is necessary to verify that there are no new holes created or no holes merged in the process of executing the redundant node deletion.

(1) If the deletion of the node v leads to the appearance of a new hole, the loop formed by the boundary edge of the new hole must be in the neighbor graph $\Gamma_G(v)$ of the node v which means that there is a loop that cannot be triangulated in $\Gamma_G(v)$. The above situation is contrary to the rule. Therefore, the deletion of the redundant node does not create new holes

(2) If the deletion of the redundant node v leads to the merging of two holes, the neighbor graph $\Gamma_G(v)$ of the node v must not be connected, which is contrary to the rule. Therefore, the deletion of the redundant node does not cause the two holes to merge

4.1. Redundant Node Sleeping. Since the border nodes are deployed at the border of the target area manually, only the internal nodes in the network execute the process of redundant node sleeping. The process of redundant node sleeping is as follows:

Step 16. Each node broadcasts two hello messages to construct its 1-simplices, 2-simplices, and 3-simplices to form the Rips complex. Each node broadcasts the first broadcast message with its ID, and every node obtains all the IDs of its 1-hop neighbors. Each node continues to broadcast the second hello message containing the IDs of its 1-hop neighbors. All nodes obtain $E(v)$ (1-simplices) when they receive their neighbor node list. $T(v)$ (2-simplices) will be obtained when those nodes receive the neighbor list of their neighbors, and 3-simplices will be formed when a neighbor node of each simplex is obtained.

Step 17. Compute k_i and C_i of the node. The internal nodes in the network are divided into two categories and obtained the determined node set V_1 and the nondetermined node set V_2 on the basis of the relationship between k_i and C_i of the node.

Step 18. Determine whether all nodes in the determined node set V_1 satisfy the redundant node determination rule. If the node v_i satisfies the rule, each determined node in $\Gamma_G(v_i)$ should be judged whether the redundant node determination rule is satisfied and whether the clustering coefficient is greater than that of node v_i; if it has, sleep the determined node v_j whose clustering coefficient is largest in $\Gamma_G(v_i)$ and satisfy the redundant node determination rule and move all neighbor nodes of node v_j out of sets V_1 and V_2, respectively. Otherwise, sleep the node v_i and move all neighbor nodes of node v_i out of sets V_1 and V_2, respectively.

Step 19. Determine whether all nodes in the nondetermined node set V_2 satisfy the redundant node determination rule. If the node v_i satisfies the rule, each nondetermined node in $\Gamma_G(v_i)$ should be determined whether the redundant node determination rule is satisfied and whether the clustering coefficient is greater than that of node v_i; if it has, sleep the nondetermined node v_j whose clustering coefficient is largest in $\Gamma_G(v_i)$ and satisfy the redundant node determination rule and move all neighbor nodes of node v_j out of V_2. Otherwise, sleep the node v_i and move all neighbor nodes of node v_i out of V_2.

Step 20. Repeat Step 17 to Step 19 until no nodes in the network need to sleep.

The deleted node which satisfies the redundant node determination rule will not affect the network structure. Simultaneously, the clustering coefficient of the node characterizes the tightness and aggregation of nodes in the network, and the larger the clustering coefficient of the node, the closer the connection between the node and the neighbors. Thus, the nodes with large clustering coefficients will be preferentially deleted without affecting the network structure.

4.2. Redundant Edge Deletion

Definition 21 (edge deletion rule). Nodes v_a, v_b, v_c, and v_d form a K_4 complete graph, and $v_a v_c$ and $v_b v_d$ are diagonal edges. At most, one edge can be deleted as a redundant edge between $v_a v_c$ and $v_b v_d$.

Proof. As shown in Figure 5(a), nodes v_a, v_b, v_c, and v_d form a K_4 complete graph with no holes where $v_a v_c$ and $v_b v_d$ are diagonal edges. No holes are generated when only the edge $v_a v_c$ or $v_b v_d$ is deleted, and the effect is shown in Figures 5(b) and 5(c). However, the hole is generated when the edges $v_a v_c$ and $v_b v_d$ are deleted simultaneously, such as in Figure 5(d).

Inference 22. Assuming that the edge $v_a v_c$ is a diagonal edge in the K_4 complete graph, the edge $v_a v_c$ can be deleted as a redundant edge if all the 2-simplices generated by the edge $v_a v_c$ are in the K_4 complete graph.

Proof. If all 2-simplices generated by the edge $v_a v_c$ are in the K_4 complete graph, then the other vertices of all 2-simplices including the edge $v_a v_c$ can generate at least one another 2-simplex; that is, no hole will be created when the edge $v_a v_c$ is deleted. As shown in Figure 6(a), the edge $v_a v_c$ is a diagonal edge of the K_4 complete graph formed by vertices v_a, v_b, v_c, and v_d. All 2-simplices generated by the edge $v_a v_c$

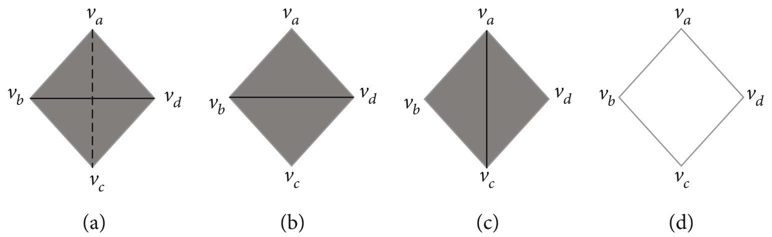


Figure 5: The proof of edge deletion rule. (a) When none of the edges are deleted (K_4), no holes exist. (b) When the edge $v_a v_c$ is deleted, no holes are generated. (c) When the edge $v_b v_d$ is deleted, no holes are generated. (d) When the edges $v_a v_c$ and $v_b v_d$ are deleted, a new hole is generated.

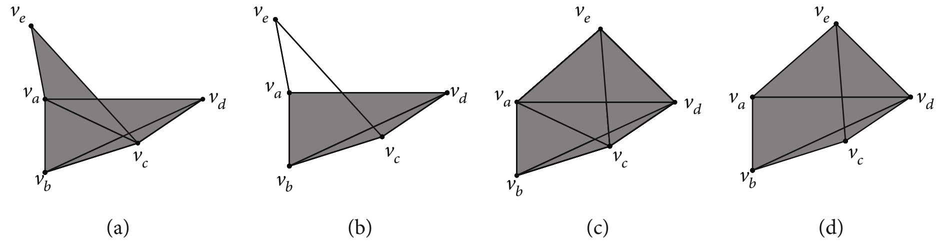


Figure 6: The proof of Inference 22. (a) The edge $v_a v_c$ is a diagonal edge in the K_4 complete graph, and not all the 2-simplices generated by the edge $v_a v_c$ are in the K_4 complete graph. (b) A new hole is generated when the edge $v_a v_c$ is deleted. (c) The edge $v_a v_c$ is a diagonal edge in the K_4 complete graph, and all the 2-simplices generated by the edge $v_a v_c$ are in the K_4 complete graph. (d) No hole will be generated when the edge $v_a v_c$ is deleted.

are $\Delta v_a v_c v_b$, $\Delta v_a v_c v_d$, $\Delta v_a v_c v_e$, $\Delta v_a v_c v_b$, and $\Delta v_a v_c v_d$, which are in the K_4 complete graph formed by vertices v_a, v_b, v_c, and v_d. Vertices v_b and v_d could also generate another 2-simplex $\Delta v_b v_d v_c$; $\Delta v_a v_c v_e$ is not in any K_4 complete graph, and the vertex v_e could not generate other 2-simplices. Then, a new hole is generated when the edge $v_a v_c$ is deleted, which can be shown in Figure 6(b) Similarly, in Figure 6(c), the edge $v_a v_c$ is a diagonal edge of the K_4 complete graph formed by vertices v_a, v_b, v_c, and v_d. All 2-simplices generated by the edge $v_a v_c$ are $\Delta v_a v_c v_b$, $\Delta v_a v_c v_d$, $\Delta v_a v_c v_e$, $\Delta v_a v_c v_b$, and $\Delta v_a v_c v_d$ which are in the K_4 complete graph formed by vertices v_a, v_b, v_c, and v_d, and vertices v_b and v_d could also generate another 2-simplex $\Delta v_b v_d v_c$. $\Delta v_a v_c v_e$ is in the K_4 complete graph formed by vertices v_a, v_c, v_d, and v_e, and the vertex v_e could also generate another 2-simplex $\Delta v_a v_d v_e$. Now, no hole will be generated when the edge $v_a v_c$ is deleted, which can be shown in Figure 6(d). In brief, Inference 22 is proven.

The process of redundant edge deletion is shown in Figure 7. First, all K_4 complete graphs are found in the network; then, the diagonal edges of each complete graph are identified and put into the queue in turn. Finally, determine whether each diagonal edge can be deleted according to Inference 22. The simplification of the Rips complex is continued if the redundant nodes exist in the network after the redundant edges are deleted. Since K_3 is 2-simplex which is not necessary to be simplified, thus the K_4 complete graphs are only considered the maximum simplicial complex.

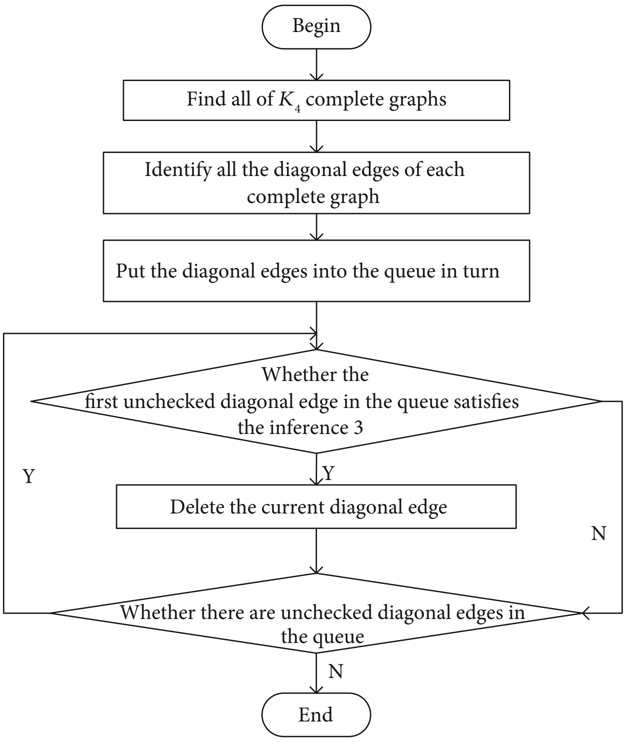

Begin
Find all of K_4 complete graphs
Identify all the diagonal edges of each complete graph
Put the diagonal edges into the queue in turn
Whether the first unchecked diagonal edge in the queue satisfies the inference 3
Y
Y
Delete the current diagonal edge
N
Whether there are unchecked diagonal edges in the queue
N
End

Figure 7: The flow chart of redundant edge deletion.

5. Hole Detection and Boundary Identification

The hole can be identified as a boundary loop composed of several boundary edges in the simplified network. Therefore, the hole identification work can be divided into the following two steps: boundary loop identification and boundary loop reduction.

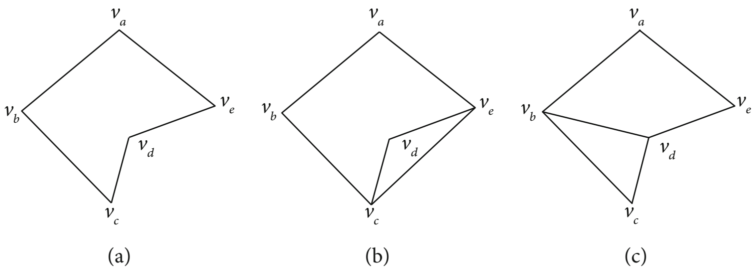


Figure 8: Boundary loop identification diagram. (a) The boundary loop with the length is 5, and v_c and v_e and v_b and v_d are neighbor nodes that are not adjacent. (b) The boundary loop with a direct connection between v_c and v_e. (c) The boundary loop with a direct connection between v_b and v_d.

5.1. Boundary Loop Identification. Finding the boundary loops is favorable to identify the boundary edges. The Rips complex is planar after the network is simplified, and a boundary edge has at most one neighbor node. Thus, the boundary edges of holes can be identified by the number of neighbor nodes of an edge in the network. For example, the number of neighbor nodes of the edge v_iv_j is the number of the same neighbor nodes of nodes v_i and v_j. It is different between internal nodes and border nodes when identifying boundary edges. Thus, the node is weighted by the case, the boundary nodes form a boundary edge whose weight is 2, and the nodes that compose the hole are called boundary nodes.

(1) The weights of two nodes are 2 if the two nodes form an edge and the number of the neighbor nodes of the edge is zero, where both nodes are border nodes

(2) The weights of two nodes are 2 if the two nodes form an edge and the number of the neighbor nodes of the edge is no more than 1, where one node is a border node and another one is an internal node

(3) The weights of two nodes are 2 if the two nodes form an edge and the number of the neighbor nodes of the edge is no more than 1, where both nodes are internal nodes

(4) Besides nodes in (1), (2), and (3), the weight of other nodes is 1

The nodes with a weight of 2 form an edge which is determined as a boundary edge. However, some edges are not the boundary edges that form a hole, and these edges are called false boundary edges which need to be deleted.

Definition 23 (false boundary edge). The boundary edge is called the false boundary edge if the neighbor nodes of this edge are distributed on the opposite side of the edge.

After the false boundary is deleted, the loop formed by the remainder boundary is the boundary loop. However, not all boundary loops are coverage holes. Therefore, it is necessary to filter the boundary loop.

Definition 24 (boundary loop filtering rule). The boundary loop will be deleted if it meets any of the following three conditions. (1) The length of the loop is 3. (2) The length of the loop is 4 and two nonadjacent nodes on the loop are neighbor nodes. (3) Two nonadjacent nodes are neighbor nodes on the loop when the length of the loop is greater than 4, and the other nodes on the loop are distributed on the opposite side of the edge composed of the two nonadjacent nodes.

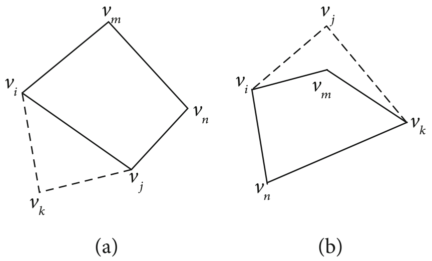


Figure 9: (a) Shortening the cycle by direct connection. (b) Shortening the cycle by node replacement.

Proof. (1) The loop is a triangle when the length of the loop is 3. Triangle holes are ignored in this study, and nontriangular holes will be identified. (2) Two nonadjacent nodes on the loop are neighbor nodes which are in a triangulated cycle when the length of the loop is 4; on the basis of the first rule, the boundary satisfying condition 2 needs to be deleted. (3) As shown in Figure 8(a), the length of the boundary loop is 5; v_c and v_e and v_b and v_d are neighbor nodes that are not adjacent. All the nodes in the cycle are distributed on the side of the connection such as in Figure 8(b); if there is a direct connection between v_c and v_e, it is uncertain that the identified loop is covered with the internal node; all the nodes in the cycle are distributed on the opposite side of the connection such as in Figure 8(c); if there is a direct connection between v_b and v_d, it is certain that the cycle is covered with a node; that is, the cycle is not the hole's boundary.

5.2. Boundary Loop Reduction. After the boundary loop is filtered, the remainder boundary loops may still not be the shortest path of the hole or several cycles contain the same hole. Therefore, it is necessary to define a loop reduction rule to shorten the boundary loop.

Definition 25 (loop reduction rule). (1) Check whether the area of the loop formed by the direct connection v_iv_j is

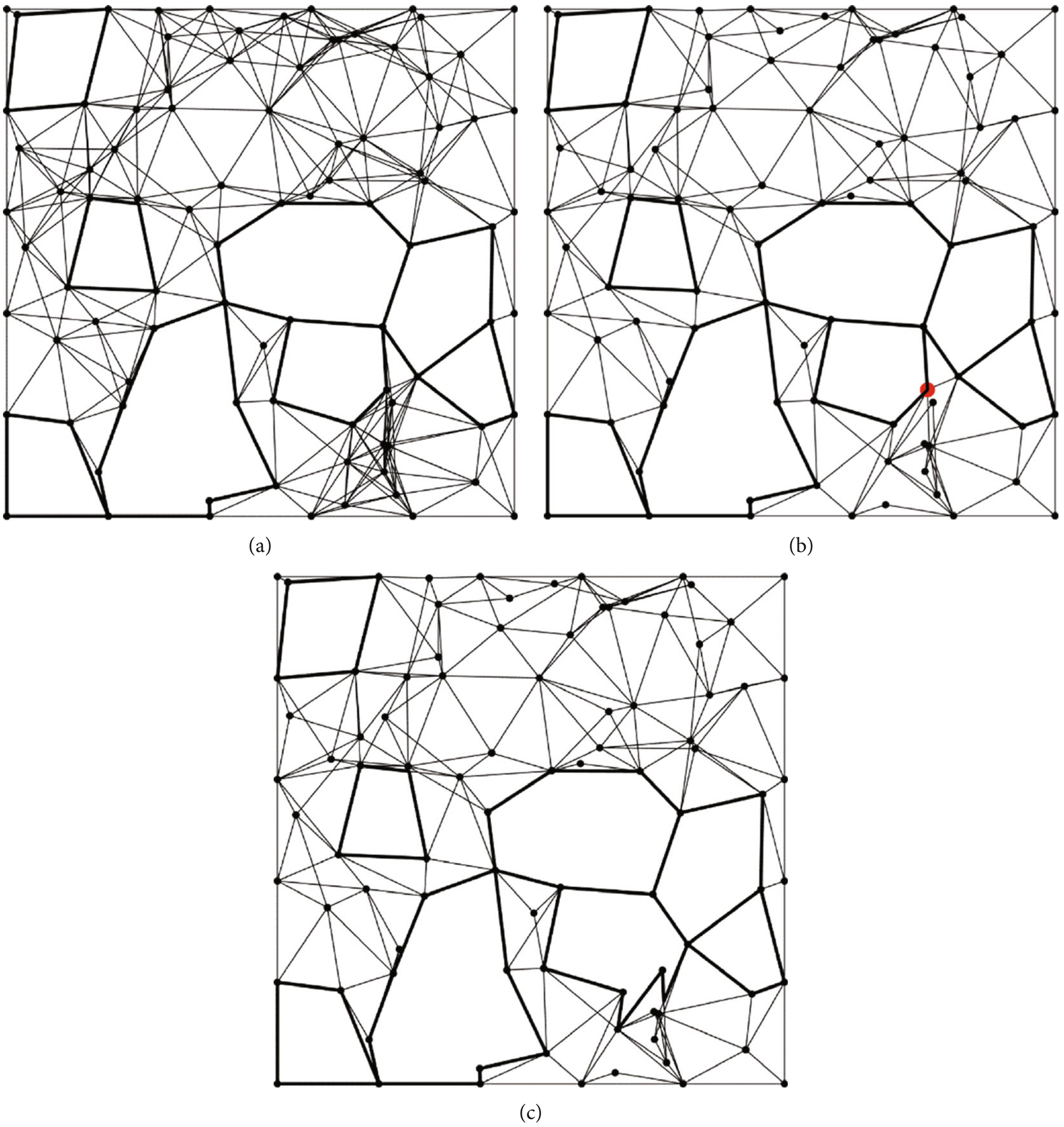


Figure 10: (a) Network diagram after the first round of redundant node sleeping. (b) Network diagram after the first round of redundant edge deletion. (c) Simplification network structure diagram.

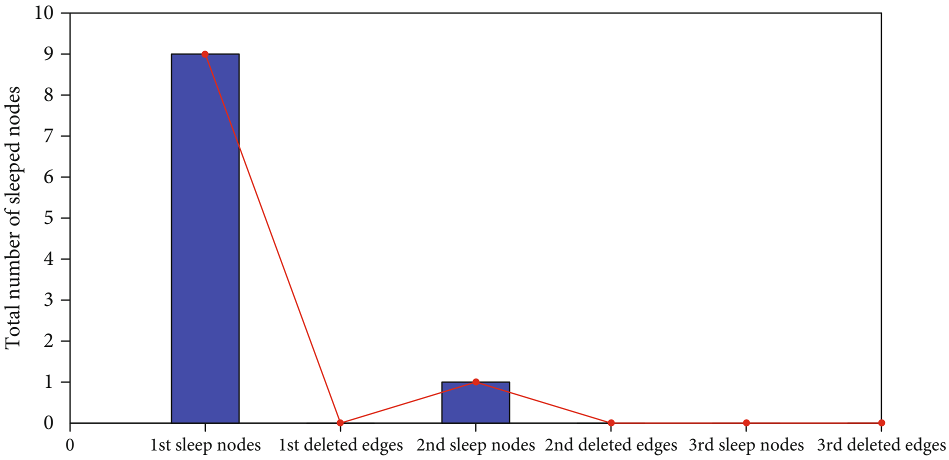
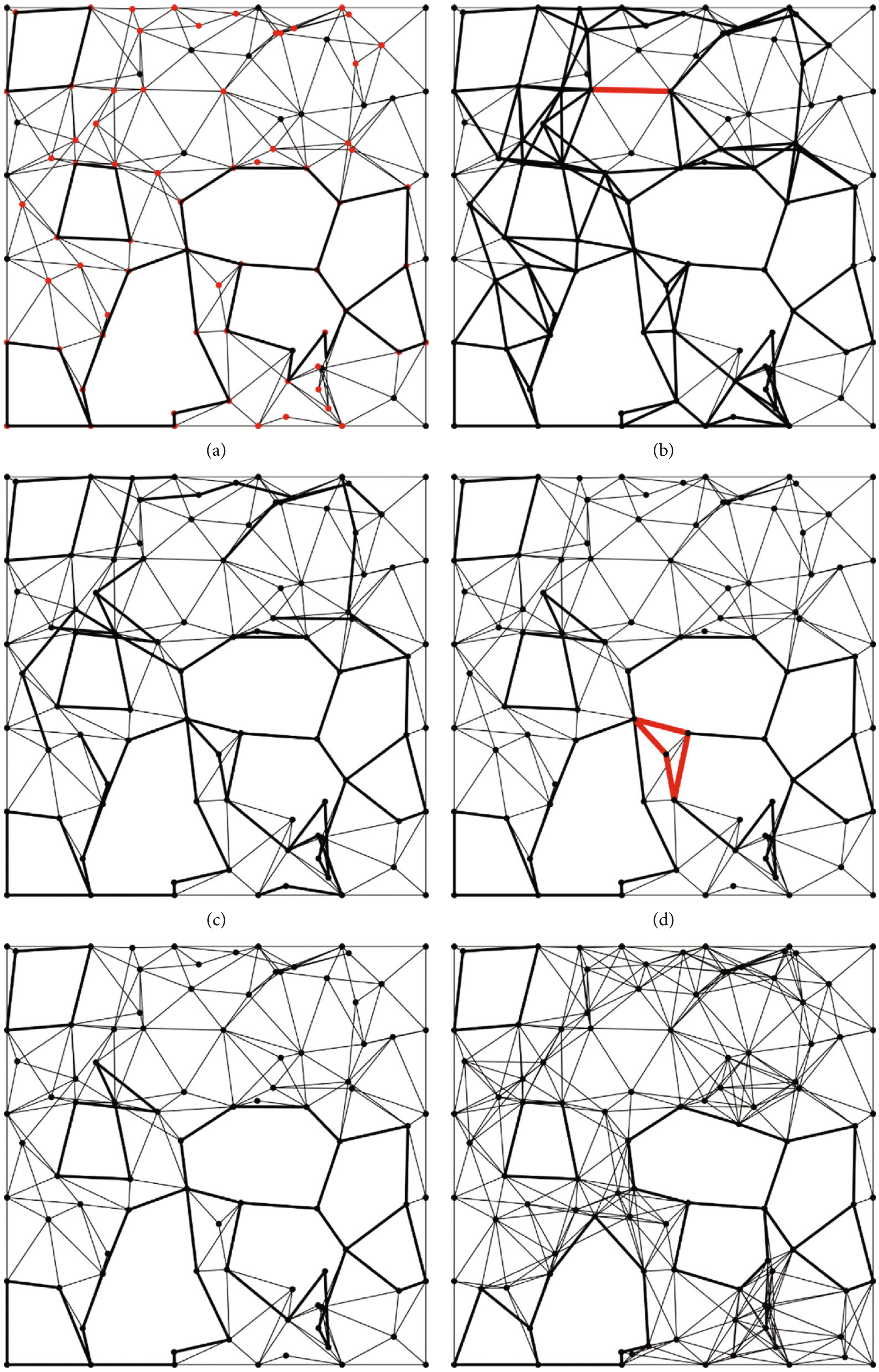


Figure 11: Node sleeping diagram. The deletion of nodes in three rounds is recorded.



(a) (b) (c) (d) (e) (f)

Figure 12: (a) Boundary node diagram (red nodes are boundary nodes). (b) Boundary edge diagram (bold lines are the boundary edges we found, and red bold line is one of the false boundary edges). (c) Network diagram after deleting false boundary edges (bold lines are the boundary edges). (d) Boundary cycle diagram (the cycle formed by bold lines is the boundary cycle we found, and the red cycle is one of the deletable cycles). (e) Filtering boundary cycle diagram (the cycle formed by bold lines is the boundary cycle). (f) Hole identification diagram (the cycle formed by bold lines is the hole boundary).

smaller than that of the original loop if the nodes v_i and v_j are neighbor nodes but not adjacent on the loop. If yes, directly connect node v_i and v_j to shorten the boundary loop, as shown in Figure 9(a). (2) v_i, v_j, and v_k are three adjacent nodes on the loop, and the node v_m is the same neighbor node of the three nodes. Check whether the area and circumference of the loop are both smaller than those of the original loop when the node v_j is replaced by the node v_m. If yes, the node v_m is used instead of the node v_j to form a new loop, as shown in Figure 9(b).

6. Simulation and Analysis

6.1. Algorithm Simulation. The hole detection accuracy and complexity are important indicators to measure the performance of the proposed algorithm and the comparison algorithms which are evaluated in this section. Simultaneously, redundant node sleeping, redundant edge deletion, and hole detection and boundary identification are further analyzed to evaluate the performance of the proposed algorithm. The parameters used in the simulation are set as follows: 100 sensor nodes are deployed in a target area of $100 \times 100\,\text{m}^2$, including 80 internal nodes randomly deployed and 20 border nodes manually deployed on the border of the target area. Each border node has two neighbor border nodes, and the distance between two adjacent border nodes is 20 m. Each sensor node has a coverage radius of 10 m and a communication radius of 20 m.

6.1.1. Redundant Node Sleeping. Sleeping redundant nodes not only simplifies the Rips complex but also is helpful for detecting the holes in the network without merging and creating new holes. Figure 1 is the original network diagram, and Figure 10(a) is the network diagram after the first round of redundant node sleeping. The rate of node sleeping is 9%. The original holes are expanded in Figure 10(a); however, no new hole merges or appears. The algorithm proposed in this paper will shorten the boundary loops later.

6.1.2. Redundant Edge Deletion. An edge deletion rule is proposed to delete the redundant edges and simplify the Rips complex without the mergence and the appearance of holes. Usually, the redundant nodes still exist in the network after the redundant edges are deleted, such as the red node in Figure 10(b); then, the simplification of the Rips complex continues. The final simplified Rips complex is shown in Figure 10(c), in which the number of holes remains but the total area increases. Generally, the simplest network structure can be achieved after two rounds of simplification and the condition of redundant node sleeping is shown in Figure 11; 10 nodes sleep after two rounds of redundant node sleeping, and one node sleeps in the second round.

6.1.3. Hole Detection and Boundary Identification. The hole detection is divided into two steps: boundary loop identification and boundary loop reduction. The proposed algorithm first finds the boundary nodes and forms the boundary edge which is composed of boundary nodes, as shown in Figure 12(a). However, some boundary edges found are not really on the boundary of holes such as the red line segment in Figure 12(b). Therefore, some boundary edges are deleted by defining the false boundary edge, as shown in Figure 12(c). But some boundary loops are not real boundaries of holes. As shown in Figure 12(d), the length of the loop with red color is 4. Two nonadjacent nodes belonging to the red loop are neighbor nodes, which meets the second rule of the redundant loop reduction rule. Therefore, some boundary loops need to be deleted according to the redundant loop rule, and the result is shown in Figure 12(e). Finally, the boundary loops are shortened as shown in Figure 12(f).

Table 1: Complexity of each step.

Step	Complexity
Redundant node sleeping	$O(n)$
Redundant edge deletion	$O(cn)$
Boundary loop identification	$O(cn)$
Boundary loop reduction	$O(mH)$

6.2. Algorithm Performance Evaluation

6.2.1. Algorithm Complexity. The detection of holes in large-scale scenarios requires lower complexity and higher efficiency. Therefore, the complexity of the algorithm is an important indicator for evaluating the efficiency of coverage hole detection. This section analyzes the complexity of the proposed algorithm. In the stage of redundant node sleeping, check whether the node meets the redundant node determination rule firstly and then check whether the clustering coefficient of the node is the largest among the neighbor nodes which can be deleted, so the algorithm complexity of this stage is $O(n)$, where n represents the number of nodes in the network. In the stage of redundant edge deletion, find all K_4 complete graphs composed of each node firstly and then judge whether all the 2-simplices composed of each diagonal edge are in the K_4 complete graph. Therefore, the complexity of this stage is $O(cn)$, where c represents the number of neighbor nodes of each edge.

The hole detection process is divided into two steps: boundary loop identification and boundary loop reduction. In the stage of boundary loop identification, the nodes are weighted to form the boundary edge firstly according to the number of neighbor nodes of an edge, and the algorithm complexity is $O(cn)$. Then, false boundary edges are deleted according to the distribution of the neighbor nodes of the boundary edge and the boundary loops are formed, so the worst complexity is less than $O(cn)$, too. Finally, in the process of filtering the boundary loop, checking whether there are two nonadjacent nodes in the loop which are neighbor nodes is needed. If it has, it is necessary to check whether other nodes in the loop are distributed in the opposite side of the line segment formed by these two nodes, so the worst complexity of the algorithm is $O(mH)$, where H is the number of the holes, m is the number of nodes in the loop, $H \ll n$, and $m \ll n$. In the stage of boundary loop reduction, check whether there are two nonadjacent nodes in the loop which

Table 2: Comparative experiment table.

Performance indicators	The proposed algorithm	HBA	Tree-based hole detection method
Algorithm complexity	$O(n)$	$O(n^3)$	$O(bn)$
Detection accuracy	99.03% (hole area)	99% (hole number)	80% (hole size)

are neighbor nodes and whether the common neighbor nodes of three adjacent nodes exist, so the worst complexity is $O(mH)$. Thus, the algorithm complexity of hole detection is $O(cn)$.

In summary, the complexity of the algorithm is $O(cn)$, and the details are shown in Table 1, where c represents the number of neighbor nodes of each edge. It is known that the HBA [22] has a complexity of $O(n^3)$ and the tree-based hole detection algorithm [12] has a complexity of $O(bn)$, where b is the number of neighbor nodes of each node. Thus, the proposed algorithm has the lowest complexity.

6.2.2. Detection Accuracy. The detection accuracy is another important indicator for evaluating the algorithm. The HBA algorithm defines the detection accuracy as the ratio of the number of detected holes to the total number of holes; meanwhile, the tree-based hole detection method defines it as the ratio of the estimated hole size to the actual hole size. Compared with the detection accuracy of the hole number, the detection accuracy of the hole area is more favorable to repair holes. Thus, the detection accuracy is defined as Equation (5) in this study.

$$r = 1 - \frac{|S' - S|}{S}, \qquad (5)$$

where S represents the actual total area of the holes, S' represents the total area of the holes detected by the proposed algorithm, and r represents the hole detection accuracy. On the basis of Equation (5), the accuracy of the proposed algorithm is 99.03%.

On the basis of the above two indicators, Table 2 concludes that the performance of the proposed algorithm is superior to that of the HBA and tree-based hole detection method.

7. Conclusions

The Rips complex constructed in this paper tends to be planar by a simplification process, which makes the coverage hole detection algorithm have low complexity and high accuracy. In the process of simplification, first, combine Turan's theorem and the concept of the degree and clustering coefficient in the complex network to define two kinds of nodes (determined nodes and nondetermined nodes) and sleep redundant nodes according to the redundant node determination rule; then, use the concept of the complete graph to derive the edge deletion rule for deleting the redundant edges. Finally, the holes in the network are detected from the perspective of the loop. Simulation results show that the detection accuracy of the hole area is 99.03% and the complexity is $O(cn)$. Detecting coverage holes in a 3-D space will be focused in our further research for detection efficiency improvement.

Data Availability

The test data, simulation data and the proposed method used to support the findings of this study are available from the corresponding author upon request. The proposed algorithm is in the state of patent-pending, thus, the access of the data used in the proposed algorithm is restricted now.

Conflicts of Interest

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

Acknowledgments

This research was funded in part by the Jilin Province Foundation for Excellent Youth Talents (grant number 20190103052JH) and Youth Foundation of Changchun University of Science and Technology (grant number QN2017013).

References

[1] K. S. Low and M. C. R. Talampas, “Wireless sensor networks for intelligent transportation applications: a survey,” in *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*, pp. 47–77, CRC Press, 2013.

[2] E. Ould-Ahmed-Vall, B. H. Ferri, and G. F. Riley, “Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, pp. 1994–2007, 2012.

[3] F. M. Al-Turjman, H. S. Hassanein, and M. A. Ibnkahla, “Efficient deployment of wireless sensor networks targeting environment monitoring applications,” *Computer Communications*, vol. 36, no. 2, pp. 135–148, 2013.

[4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, “Coverage problems in wireless ad-hoc sensor networks,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, pp. 1380–1387, Anchorage, AK, USA, 2001.

[5] X. Deng, Y. Jiang, L. T. Yang, M. Lin, L. Yi, and M. Wang, “Data fusion based coverage optimization in heterogeneous sensor networks: a survey,” *Information Fusion*, vol. 52, pp. 90–105, 2019.

[6] R. Tan, G. Xing, B. Liu, J. Wang, and X. Jia, “Exploiting data fusion to improve the coverage of wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 450–462, 2012.

[7] I. Khan, H. Mokhtar, and H. Merabti, “An overview of holes in wireless sensor network,” in *Proceedings of the 11th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, 2010.

[8] P. Corke, R. Peterson, and D. Rus, “Finding holes in sensor networks,” in *Proceedings of the IEEE workshop on omniscient space: robot control*, 2007.

[9] F. Li, J. Zheng, and B. Zhang, “Geographic hole-bypassing forwarding protocol for wireless sensor networks,” *IET Communications*, vol. 5, no. 6, pp. 737–744, 2011.

[10] T.-W. Sung and C.-S. Yang, “Voronoi-based coverage improvement approach for wireless directional sensor networks,” *Journal of Network and Computer Applications*, vol. 39, no. 1, pp. 202–213, 2014.

[11] 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.

[12] W. Li and Y. Wu, “Tree-based coverage hole detection and healing method in wireless sensor networks,” *Computer Networks*, vol. 103, pp. 33–43, 2016.

[13] A. Soundarya and V. Santhi, “An efficient algorithm for coverage hole detection and healing in wireless sensor networks,” in *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech)*, pp. 1–5, Kolkata, India, 2017.

[14] P. Kumar Sahoo, M.-J. Chiang, and S.-L. Wu, “An efficient distributed coverage hole detection protocol for wireless sensor networks,” *Sensors*, vol. 16, no. 3, p. 386, 2016.

[15] Y. Bejerano, “Coverage verification without location information,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 631–643, 2012.

[16] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, “Lifetime and coverage guarantees through distributed coordinate-free sensor activation,” *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 470–483, 2011.

[17] P. J. Si, C. D. Wu, Y. Z. Zhang, Z. Xia, and L. Cheng, “A hole detection and redeployment strategy in wireless sensor network,” *Journal of Information and Computational Science*, vol. 8, no. 13, pp. 2577–2585, 2011.

[18] X. M. Zhao, K. J. Mao, F. Yang, W. Wang, and Q. Chen, “Research on detecting sensing coverage hole algorithm based on OGDC for wireless sensor networks,” *The Journal of Computer Information Systems*, vol. 8, no. 20, pp. 8561–8568, 2012.

[19] J. W. Zhang, S. W. Li, Q. Y. Li, Y. C. Liu, and N. N. Wu, “Coverage hole recovery algorithm based on perceived probability in heterogenous wireless sensor network,” *The Journal of Computer Information Systems*, vol. 10, no. 7, pp. 2983–2990, 2014.

[20] A. Tahbaz-Salehi and A. Jadbabaie, “Distributed coverage verification in sensor networks without location information,” *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1837–1849, 2010.

[21] F. Yan, P. Martins, and L. Decreusefond, “Connectivity-based distributed coverage hole detection in wireless sensor networks,” in *IEEE Global Telecommunications Conference - GLOBECOM 2011*, pp. 1–6, Houston, TX, USA, 2011.

[22] F. Yan, A. Vergne, P. Martins, and L. Decreusefond, “Homology-based distributed coverage hole detection in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1705–1718, 2015.

[23] G. Wang, G. Cao, P. Berman, and T. F. la Porta, “Bidding protocols for deploying mobile sensors,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 563–576, 2007.

[24] Y. Zhang, X. Zhang, Z. Wang, and H. Liu, “Virtual edge based coverage hole detection algorithm in wireless sensor networks,” in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1488–1492, Shanghai, China, 2013.

[25] C. Qiu and H. Shen, “A Delaunay-based coordinate-free mechanism for full coverage in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 828–839, 2014.

[26] S. P. Fekete, A. Kroeller, D. Pfisterer, S. Fischer, and C. Buschmann, “Neighborhood-based topology recognition in sensor networks,” *Computer Science*, vol. 3121, no. 2, pp. 18-19, 2004.

[27] S. Funke and C. Klein, “Hole detection or: how much geometry hides in connectivity?,” in *Proceedings of the twenty-second annual symposium on Computational geometry - SCG '06*, pp. 377–385, Sedona, AZ,USA, 2006.

[28] X. Wang, S. Han, Y. Wu, and X. Wang, “Coverage and energy consumption control in mobile heterogeneous wireless sensor networks,” *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 975–988, 2013.

[29] A. Muhammad and M. Egerstedt, “Control using higher order Laplacians in network topologies,” in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, pp. 1024–1038, Kyoto, Japan, 2006.

[30] B. Wang, X. Deng, W. Liu, L. Yang, and H. C. Chao, “Confident information coverage in sensor networks for field reconstruction,” *IEEE Wireless Communications*, vol. 20, no. 6, pp. 74–81, 2013.

[31] L. Yi, X. Deng, M. Wang, D. Ding, and Y. Wang, “Localized confident information coverage hole detection in Internet of things for radioactive pollution monitoring,” *IEEE Access*, vol. 5, pp. 18665–18674, 2017.

[32] L. Yi, X. Deng, Z. Zou, D. Ding, and L. T. Yang, “Confident information coverage hole detection in sensor networks for uranium tailing monitoring,” *Journal of Parallel and Distributed Computing*, vol. 118, Part 1, pp. 57–66, 2018.

[33] X. Deng, L. T. Yang, L. Yi, M. Wang, and Z. Zhu, “Detecting confident information coverage holes in industrial Internet of things: an energy-efficient perspective,” *IEEE Communications Magazine*, vol. 56, no. 9, pp. 68–73, 2018.