Journal of Ambient Intelligence and Humanized Computing (2023) 14:5873–5884
https://doi.org/10.1007/s12652-020-02038-3

ORIGINAL RESEARCH

CHPT: an improved coverage-hole patching technique based on tree-center in wireless sensor networks

Smita Das[1] · Mrinal Kanti Debbarma[1]

Received: 2 December 2019 / Accepted: 24 April 2020 / Published online: 25 May 2020

Abstract

The occurrence of coverage-hole (CH) in a wireless sensor network degrades the performance of the network in terms of data transmission. Therefore, after the detection of CH, hole-patching is the next important task in order to reduce the energy consumption and to improve the network lifetime. In this paper, we propose an improved coverage-hole patching technique (CHPT) based on the concept of center of tree. The prerequisite to hole-patching is the hole-detection, which is done based on Delaunay Triangle and empty circle property. Subsequently, using the Inner empty circle property, the hole-area is estimated. And finally, CHPT is used to determine the hole-boundary and formulate the sub-trees for identifying the patch-positions to be covered by deploying additional sensor nodes. Experimental results show that the average percentage of coverage in CHPT increases up to 98.6% in terms of additional sensors and with a given sensing radius.

Keywords Coverage-hole · Hole-area estimation · Hole-boundary detection · Hole-healing · Patch-positions · Wireless sensor networks

1 Introduction

Now-a-days, WSNs have massive application in various fields which are designed in such a way that the target object can be monitored without interruption. Habitat monitoring (Kar and Banerjee 2003), traffic monitoring (Roman et al. 2006), air pollution monitoring (Ratnasamy et al. 2003), agricultural monitoring (Khan and Kumar 2019), medical diagnosis (AbdulGhaffar et al. 2020) etc. are some of the major applications which are based on monitoring the area (Das and Debbarma 2019b). With respect to the conventional network, the basic unit of a wireless sensor network is the cost-effective sensor node (SN) that needs to be energy efficient to gather data from some required environment. Although these SNs have limited processing speed and memory, still they are proficient enough to sense, evaluate and collect data from the environment. The random deployment of SNs in the Region of Interest (RoI) is the main reason behind the creation of CH. The existence of CH in the RoI can cause a number of problems (Das and DebBarma 2018) such as: deteriorated performance of sensor network, improper communication between the nodes, decreased network lifetime and many more. Therefore, to improve the performance of the network, detection of CH becomes an important research topic. In this connection, various papers (Liang et al. 2014; Zhang et al. 2016) further converse about the hole-patching issues. Coverage-hole patching is the technique of deploying sensor nodes on the identified positions, known as *Patch Positions*, so that they are covered partially or completely by the deployed sensor nodes. In this proposed work, our foremost goal is to patch or heal the coverage-holes by deploying sensor nodes on the identified patch positions to improve the *coverage ratio* which is the ratio of coverage-area before and after hole-healing. Based on hole-recovery, the coverage-holes can be classified into two categories (Le Nguyen et al. 2019)

- Patchable-hole: They are created by adversarial attacks or exhaustion of battery.This type of coverage-holes can be patched by deploying new sensors.

✉ Smita Das
smitadas.nita@gmail.com

Mrinal Kanti Debbarma
mkdb06@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology Agartala, Jirania, Tripura 799046, India

- Unpatchable-hole: They are created due to the presence of large, immovable obstacles such as rivers, buildings etc. This second type of hole cannot be patched at all.

In this work, we have considered only the patchable-holes so as to design the CHPT algorithm in such a way that CH gets patched as soon as they get detected by deploying new nodes at suitable patch positions.

The rest of the paper is organized as follows: Sect. 2 describes our motivation behind choosing this research problem and the innovative solutions for the problem. Section 3 depicts the related work with the background of CH patching. In Sect. 4, the proposed method is discussed elaborately. Section 5 represents simulation and result analysis. Finally, we have concluded the paper in Sect. 6.

2 Motivation and innovation

In present day, WSN is considered as one of the main enabling technologies for Internet of Things (IoT). Therefore, enhancing the node deployment and coverage services of WSNs has a direct impact on the realization of IoT. The main challenge in such IoT setting is to determine the coverage-holes if any. Non-deterministic sensing node deployment often makes CHs inevitable even in a high-density network. If there is some large coverage-hole in the network, then communication becomes weak between the nodes due to the fact that, sensed data is routed along the boundary of the hole repeatedly. Therefore, detection of CH is very essential because it enhances the network lifetime, provides sufficient information about the coverage of each point in the RoI and also determines the requirement of extra nodes for speedy coverage.

Therefore, let us assume that in a given 2D RoI, a set of static, homogeneous sensors S are placed randomly. Let, each sensor has a communication radius as R_C and a sensing radius as R_S where the relationship of R_C and R_S can be arbitrary. For coverage, we have considered the binary sensing models under which, a location can be monitored by a sensor if it is within the sensor's sensing radius. Thus, the proposed work can be defined more precisely as follows: A number of static sensors $S = [s_1;s_2;\ldots;s_n]$ will be deployed randomly in the RoI due to which a number of coverage holes $C = [c_1;c_2;\ldots;c_n]$ get created and are identified. In this paper, we have considered the holes inside the network only and ignored the open-holes. The hole-area will be estimated based on the area of each coverage-hole. Therefore, hole-area is being calculated as: $A_r = [a_1;a_2;\ldots;a_n]$. Thereafter, the hole-boundary will be detected for the patch-position identification. Finally, additional, static sensor nodes will be deployed to the identified patch positions $P = [p_1;p_2;\ldots;p_n]$ such that maximum uncovered hole-area is patched.

Hence, in this paper, our main contributions are: (1) identification of CH boundary: after CH identification based on node-positioning and Delaunay Triangulation (DT), tree-nodes are used to represent the CH for identification of hole-boundary. (2) Sub-tree formation: after hole-merging, the actual hole-area is identified based on inner empty circle (IEC) property. Subsequently, the whole CH is again divided into sub-holes for easy identification of patch positions. (3) Patch-position identification: based on tree-centers, patch-positions are identified to be patched by additional node deployment.

3 Related work and background

This section provides more insight about the background of hole-patching issues and its prerequisites which are discussed below.

3.1 Coverage-hole detection

Detection of CH confirms whether extra nodes are required to fill up those CHs or not. The authors in Li and Hunter (2008) and Kanno et al. (2009) proposed distributed algorithms to detect the existence of CHs and also recovered them. Yang and Fei (2010) proposed a hole-detection and adaptive geographical routing (HDAR) algorithm to detect CHs and to deal with local minimum problem. Yan et al. (2011) used the concepts of Rips complex and Cech complex to discover coverage holes and classify CHs to be triangular or non-triangular. Beghdad and Lamraoui (2016) proposed an algorithm based on connected independent sets (BDCIS) for selection of internal node or boundary nodes based on which CHs get identified. In our proposed method, before hole-detection, the position of the nodes has to be determined. Practically, while positioning the nodes, errors may exist. Due to positioning errors (Ghaffari 2019), many covered regions can be identified as holes or vice versa. Therefore, based on the experiments done in Das and Debbarma (2019a), we have identified the node positions using which the network topology is represented using Delaunay Triangle. Delaunay Triangle is formed with nodes in such a way that the circumcircle of every triangle is empty. Again, DT possesses the empty circle property which says that the circumcircle of DT is the special circle that crosses through the three vertices of the DT. An EC is shown in Fig. 1. Now, as per Li and Zhang (2015) a coverage-hole exists in the RoI if the communication radius is greater than the sensing radius i.e., $R_C > R_S$. That means, if radius of EC is larger, definitely the center of the EC is out of coverage of sensing range of any sensor node. Thus, there must be some uncovered region

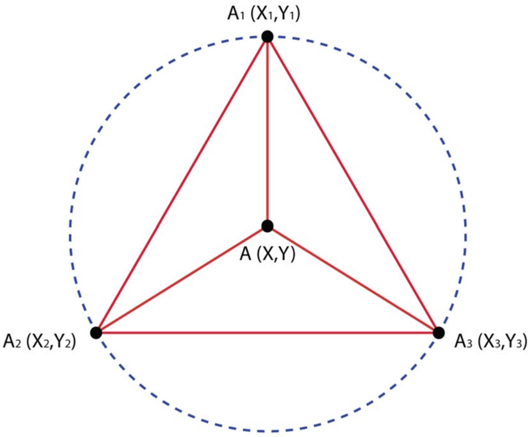


Fig. 1 Empty circle representation

in the EC that contains its center. As per definition, this uncovered region itself is the CH. Hence, proved that for $R_C > R_S$, there is some uncovered region in the empty circle.

3.2 Coverage-hole merging

Based on hole-detection, several local holes may be identified in the RoI with respect to certain random node arrangements (Musa et al. 2019). There may be some situation where two or more CHs can be merged to obtain a single hole. Therefore, for attaining the global view of such CHs, a hole-merging technique has to be followed. In this paper, CH merging is done with the IEC concept used in Li and Wu (2016). Advancement of empty circle is the inner empty circle (IEC) which is a concentric circle with its corresponding EC. If R_E is the radius of an EC and $R_E > R_S$, then radius of IEC, $R_{IEC} = (R_E - R_S)$. In Fig. 2, red circles are showing the ECs and the green circles represent the IECs. If any two separate IECs are generated from two ECs of two adjacent DTs, then those IECs must have a common side. Again, If the connector between the centers of those two IECs intersects the common part of the two DTs and the length of the common side is greater than (2*sensing radius), then the two IECs intersect the same CH. Thus, the separated IECs can be merged to provide a global view of the CHs.

From Fig. 2, we can explain the hole-merging concept as: $\Delta A_1A_2A_3$ and $\Delta A_1A_4A_3$ are the two neighboring DTs. The red dotted circles denote the ECs and the green dotted circles denote the IECs. The line segment C_1C_2 joins the centers of the two IECs of the two DTs. Here C_1C_2 intersects the common side A_1A_3 of the two DTs. Again, from Fig. 2, R_S being the sensing radius, length of $A_1A_3 > (2 * R_S)$. If we take any point on C_1C_2 then its distance from any vertices of $\Delta A_1A_2A_3$ and $\Delta A_1A_4A_3$ is greater than R_S. Further, it can be found that no point on C_1C_2 is covered by any sensor and C_1 and C_2 are from different IECs. Thus, the uncovered regions in the two IECs belong to the same CH.

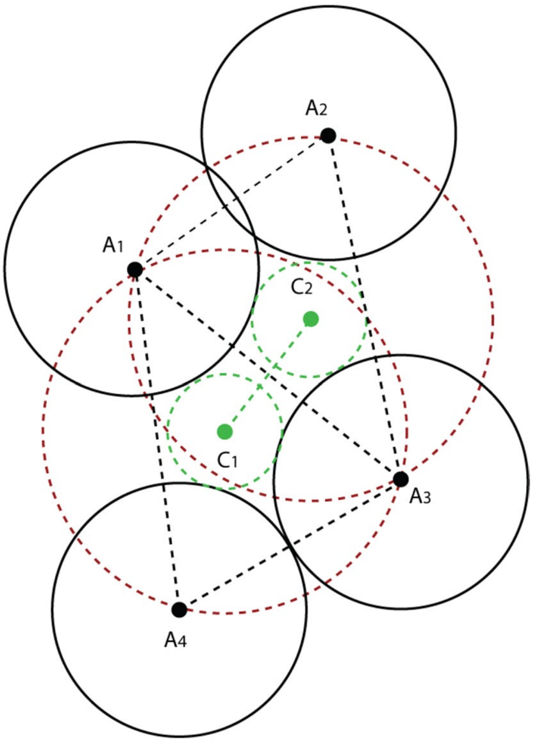


Fig. 2 Inner empty circle representation

3.3 Hole area estimation

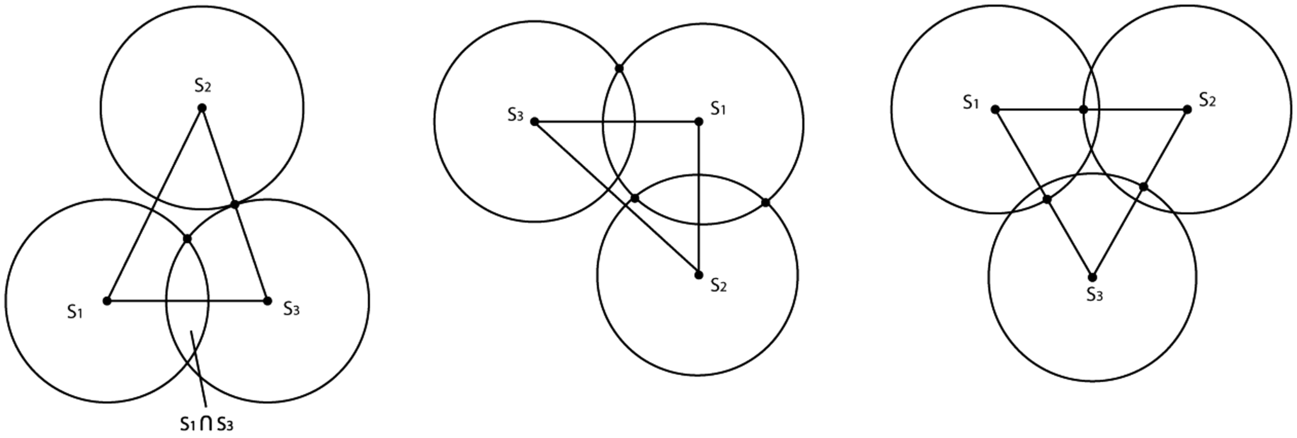
After identifying coverage-holes and merging them, the requirement is to find the area of such coverage-holes. In this paper, hole-area estimation is done with the IEC concept. EC can identify the hole but the size of EC and hole may not be same. Since $R_E > R_S$, the size of an EC can be much larger than the subsequent coverage-hole and that is why IEC is used for calculating the hole-area much accurately. Das and Kanti DebBarma (2018) proposed the hole-area estimation with one, two or three intersections using computational geometry based approach which is discussed briefly as follows:

Let us assume the three sides of the Delaunay Triangle are: a, b and c respectively and there are three situations with one, two and three intersections as shown in Fig. 3. Now, area of triangle, $A_t = \frac{1}{4}\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)}$, where $s = \frac{(a+b+c)}{2}$

Figure 3a shows area-estimation with one intersection.

Area of hole, $A_h = A_t - (A_1 + A_2 + A_3) - (S_1 \cap S_2)$

where A_1, A_2 and A_3 are the area of each circular sector with respect to the sensor nodes S_1, S_2 and S_3.



(a) One Intersection **(b)** Two Intersections **(c)** Three Intersections

Fig. 3 Hole-area with one, two and three intersections

Again, $A_1 + A_2 + A_3 = \pi R_S^2$ where R_S is the sensing radius of each sensor node by assuming $R_a = R_b = R_S$ (since each sensor have a fixed sensing radius of R_S). Therefore, with one intersection, area of CH,

$$A_h = \frac{1}{4}\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)} \\ - \pi R_S^2 - 2R_S^2 \cdot cos^{-1}1\left(\frac{d}{2R_S}\right) - \frac{d}{2} \cdot \sqrt{4R_S^2 - d^2}$$

where d is the distance between two sensor nodes for one intersection.

Similarly, Fig. 3b depicts area-estimation with two intersections. Here area of CH,

$$A_h = \frac{1}{4}\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)} \\ - \pi R_S^2 - 2R_S^2 \cdot \left[cos^{-1}1\left(\frac{d_1}{2R_S}\right) + cos^{-1}1\left(\frac{d_2}{2R_S}\right)\right] \\ - \frac{1}{2} \cdot \left[d_1\sqrt{4R_S^2 - d_1^2} + d_2\sqrt{4R_S^2 - d_2^2}\right]$$

where d_1 and d_2 are the distances between nodes for two intersections.

And finally, area-estimation with three intersections is shown in Fig. 3c, when area of CH,

$$A_h = \frac{1}{4}\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)} \\ - \pi R_S^2 - 2R_S^2 \cdot \left[cos^{-1}1\left(\frac{d_1}{2R_S}\right) + cos^{-1}1\left(\frac{d_2}{2R_S}\right) + cos^{-1}1\left(\frac{d_3}{2R_S}\right)\right] \\ - \frac{1}{2} \cdot \left[d_1\sqrt{4R_S^2 - d_1^2} + d_2\sqrt{4R_S^2 - d_2^2} + d_3\sqrt{4R_S^2 - d_3^2}\right]$$

where d_1 and d_2 and d_3 are the distances between nodes for three intersections. Hole-area is estimated before and after hole-patching and compared to identify the coverage ratio in order to check for coverage improvement.

3.4 Hole-boundary detection methods

In a WSN, the hole-boundary nodes enclose the region in the ROI where sensing or transmitting is not possible. Generally, hole-boundary is composed of the nodes which are out of functions (Yu et al. 2001) and the various methods for hole-boundary detection are as follows:

Kang et al. (2013) proposed a connectivity-based, distributed algorithm based on boundary critical point (BCP) and can run on a single node by verifying the BCPs from neighbors. Ghosh et al. (2014) proposed two novel distributed hole-detection algorithms as DVHD (distance vector hole-determination) and GCHD (Gaussian curvature based hole-determination). Using distributed Delaunay Triangulation, DVHD identifies the hole-boundaries. Jing et al. (2014) in the same year proposed a geometric-topologic hybrid method for boundary detection suitable for large-scale holes based on minimum critical threshold constraint (BLW-MCT). Li and Zhang (2015) proposed a distributed algorithm for hole-detection based on DT and empty circle property. Along with hole-detection, their proposed algorithm also identified the boundary nodes based on node clustering. Zhang et al. (2017) proposed computational geometry based approach for hole-detection and boundary identification in a hybrid sensor network containing static as well as mobile sensors. The summary of the above literature can be tabulated in Table 1 as following:

3.5 Various hole-patching algorithms

After patch-position identification, the number of nodes required to fill up those positions may be calculated in order to enhance the coverage quality. The various hole-patching algorithms are basically of three types-*Topology based, Computational Geometry based* and *Statistical approach based*. Topology based approach is using the various topological properties like connectivity information to identify the boundary nodes. They do not need any extra location identification device such as GPS and hence cost is minimized. But they have to collect the neighboring nodes information and therefore high control packet overhead may occur. Still they have moderate accuracy w.r.t. statistical methods. Computational Geometry based method is using node location information. They may collect the node location either by using GPS or to avoid higher cost, geometry based node localization methods may be used. Computational geometry (CG) based approach is much accurate in comparison to topological and statistical approaches. In case of statistical approach, sensor nodes are assumed to be distributed uniformly in the ROI. There is no requirement of node location information in this case. Therefore, based on statistical property such as: probability of node distribution and under a specific network model, boundary nodes can be identified. This approach has less accuracy in comparison to topology and CG based approaches. After comparing all those approaches, it is found that the computational geometry based approach is the most suitable, since it is most cost-effective and accurate. The various algorithms for hole-patching are as follows:

Wang and Tseng (2008) proposed a centralized method for hole-patching which is deterministic and is used to achieve a multilevel coverage. Authors claimed that due to interpolating placement scheme, less number of nodes is required to ensure k-coverage of the RoI using static nodes. Shen et al. (2010) used a bipartite graph concept to address the mobile sensor redeployment problem. By partitioning the area of a WSN into a number of grids, the gap of each grid is filled up using optimum mobile nodes by minimizing the total movement cost of all mobile sensors. Liu (2012) proposed an ant colony optimization with three classes of ant transitions, for the problem of minimum-cost and connectivity guaranteed grid coverage (MCGC). Kang et al. (2013) proposed a distributed, node-based coverage hole detection algorithm based on boundary critical point (BCP). Also, it can run on a single node with verifying hole-boundary and the hole-patching implemented with the concept of perpendicular bisector. Mellouk et al. (2013) proposed a distributed, virtual forces-based local healing approach in a mobile WSN where a high node density is required. The SNs those are located at an appropriate distance from the hole are considered for the healing process. In 2016, in a benchmark algorithm by Li and Wu (2016) proposed a tree-based method to identify the optimal patch positions for hole-healing by hole-dissections. The tree is able to indicate the locations, sizes and shapes of CHs precisely. In more recent works, Zhang et al. (2019) proposed an online cascade hole healing problem in which holes appear online and should be healed immediately without knowing future locations of holes. Two targets are considered, aiming to minimize the total energy consumption and the largest individual energy consumption, respectively. Le Nguyen et al. (2019) proposed a novel protocol TELPAC which efficiently locate the hole-boundary and determine the patching locations. The main idea behind TELPAC is to approximate the hole by a polygon whose edges are aligned by a regular triangle lattice. In the same year, Yan et al. (2020) proposed a robust approach based on an improved artificial fish swarm algorithm. The movement of mobile nodes is analysed to the motion of artificial fish with the network coverage rate as objective function. Singh and Chen (2019) proposed a chord based hole-detection method, for identifying both closed and open hole in the RoI. Their proposed chord based hole covering method provides complete sensing coverage of the network using minimum number of

Table 1 Summarization of hole-boundary detection using CG based approach

Ref.	Key features	Disadvantage	Future work
Kang et al. (2013)	homogeneous network, RC = 2RS	High computational complexity	Real life application
Ghosh et al. (2014)	Construction of DT, Incident angle	High communication overhead	Compliance to link failure
Jing et al. (2014)	Large scale hole detection, MCT constraints	–	–
Li and Zhang (2015)	Node clustering, hole and network boundary detection	Cannot detect boundary holes	Minimum cost hole-healing
Zhang et al. (2017)	Auxiliary node deployment	Cannot detect boundary holes	Hole detection in directional network

sensors. The main pros and cons of various healing methods discussed here are summarized in Table 2.

4 Proposed hole-patching technique

This is the main part of CHPT algorithm. In this algorithm, the CH is represented using a tree. For understanding the algorithm in a better way, the following definitions are required:

Tree: A tree is an undirected graph which is acyclic and there exist exactly one path in between every pair of vertices.

Distance: The distance between any two vertices (say v_i and v_j) in a tree is the length of the only one path between them.

Eccentricity: The eccentricity of a vertex v in a tree is the distance from v to the vertex farthest from v in the tree.

Center: The vertex with minimum eccentricity value is called the center of the tree.

Pendant Node: A vertex of a tree is known to be pendant if its neighborhood contains exactly one vertex.

1-D Neighbour: The neighbouring nodes which are at distance 1, from the center or any other node of the tree are known as 1-D Neighbour of the tree.

The main three steps in the CHPT are: (a) identification of hole-boundary, (b) sub-tree formation and (c) patch-position identification

4.1 Identification of hole-boundary

Generally, hole-boundary is formulated by the dead nodes those are inactive. Hence, hole-boundary detection is very important for accurate hole-area calculation and to estimate the number of extra node required for hole-healing. From Fig. 4, it can be found that the CH is represented using a tree such that the centers of each IEC (shown in red circles) acts as a node in the tree. For this, from the tree, we have to find out the pendant nodes, i.e. the nodes having exactly one adjacent node. For example, nodes 1, 5, 9, 10, 11 and 12 are the pendant nodes in Fig. 4. Now, the SNs associated with those IECs are connected for forming the actual hole-boundary and is shown in Fig. 5.

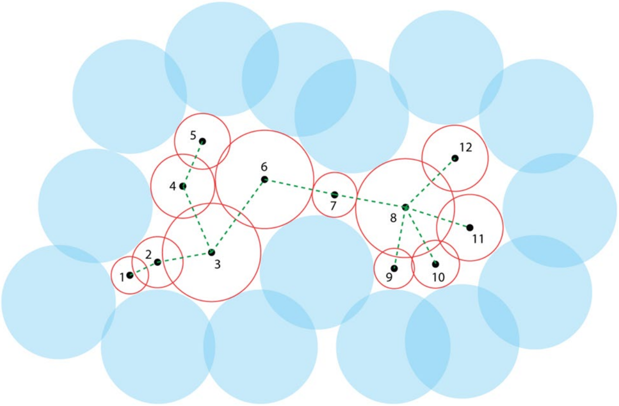


Fig. 4 Tree based hole-representation

4.2 Formation of sub-trees

Sub-trees are required to be formed for better identification of the patch-positions. The tree shown in Fig. 5 has 12 nodes in it. The centers of each of the node is denoted by the numbers 1–12. From Fig. 5, we can determine the center of T based on eccentricity value. Eccentricity is denoted by $E(v)$ and it is the distance between the node v and the farthest node in T where $v \in T$. Figure 6 shows the tree along with eccentricity value (denoted in red) for each node.

Table 2 Summarization of hole-patching using CG based approach

Ref.	Key features	Pros	Cons
Wang and Tseng (2008)	Deterministic, autonomous mobile sensors	Multilevel coverage	High communication overhead
Shen et al. (2010)	Bi-partite graph, mobile sensor redeployment	Minimizes gap between grids	High computational complexity
Liu (2012)	Minimum-cost and connectivity guaranteed grid coverage	Better searching speed	Full coverage is not guaranteed
Kang et al. (2013)	Boundary critical point based	–	Cannot provide full coverage
Mellouk et al. (2013)	Virtual force based	Local healing	Cannot detect boundary holes
Li and Wu (2016)	Tree based	Hole-dissection	Hole detection in directional network
Zhang et al. (2019)	Online hole-healing	Location of hole may be unknown	–
Le Nguyen et al. (2019)	Hole-boundary detection	Efficient patch positioning	Optimizing the number of patching sensors
Yan et al. (2020)	Hybrid WSN, hole-recovery	Objective function as coverage rate	–
Singh and Chen (2019)	Chord based	identify both open and close holes	–

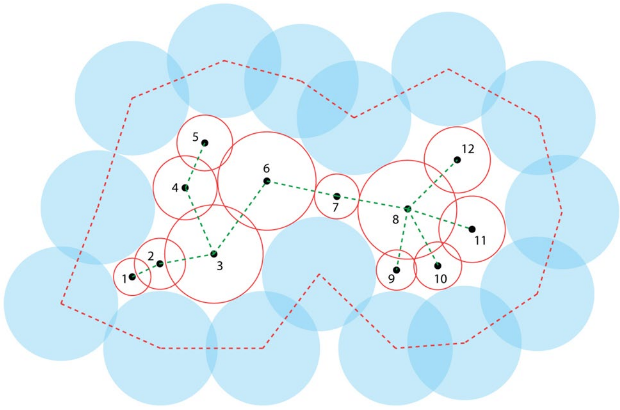


Fig. 5 Hole-boundary representation

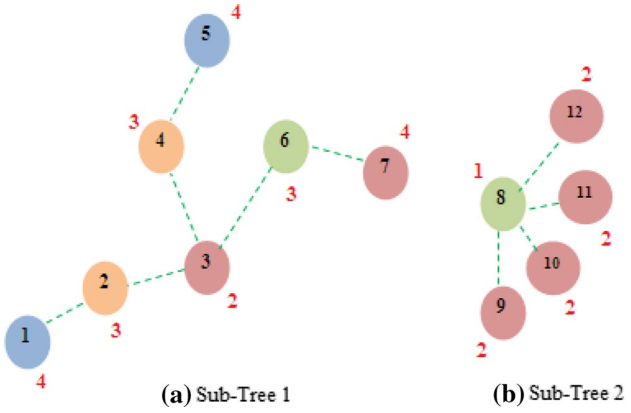


Fig. 8 Center identification from sub-trees

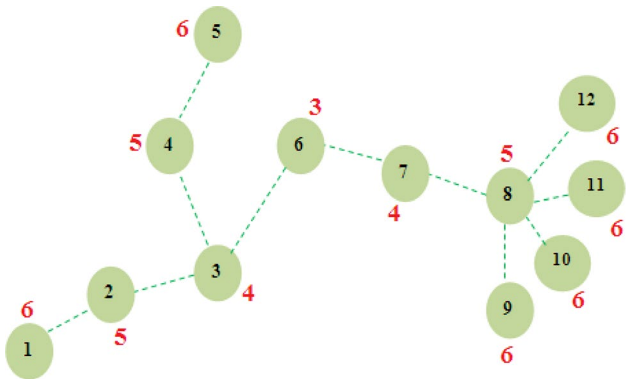


Fig. 6 Eccentricity value calculation

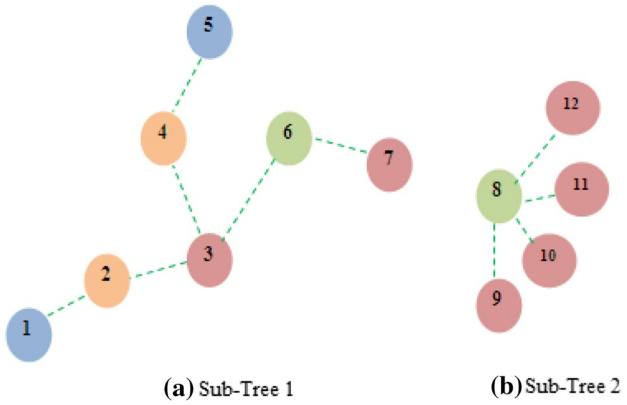


Fig. 7 Sub-tree formation

After calculating the $E(v)$ for each node, it is found that node 6 possesses the lowest $E(v)$ which is 3. Since, the node with lowest $E(v)$ is termed as the center of the tree, here node 6 is the center of the tree T. Figure 7 shows the sub-tree formation in details. After identifying the center of the tree, we denote it as the root (coloured in green) of the sub-tree to be formed. Now we add the two 1-D (at distance 1 from center) neighboring nodes 3 and 7 (coloured in red) with the root 6 to form one sub-tree. Subsequently, we check for the remaining nodes (IECs) which are smaller in size (as per Fig. 5) than the 1-D neighbors. If smaller nodes are available, they will be connected with the previously formed sub-tree or else a separate sub-tree will be formed.

Therefore, from Fig. 7, the root node is 6 (green coloured) for the first sub-tree and 3 and 7 are it's 1-D neighbors (red coloured). Again, w.r.t. 3, its 1-D neighbors are 2 and 4 (orange coloured) which are smaller in size. Subsequently, 1 and 5 (blue coloured) are the 1-D neighbors of 2 and 4, respectively. On the contrary, node 8 is larger in size than 7 and thus it is not included in the previous sub-tree. Rather, node 8 will be the root for the next sub-tree to be formed with the 1-D neighbors 9, 10, 11 and 12. The reason behind the creation of sub-tree from the main tree is that the whole CH is now divided into two small holes so that it becomes easier for us to identify the patch positions and to deploy sensor nodes required to heal them. The main motive behind the proposed work is to identify the patch positions to be filled using extra sensor nodes and to provide better coverage-rate. During the hole-boundary detection, it is found that due to random deployment, the shape of the coverage-hole may be uneven. In addition, it is very difficult to cover-up the uneven shaped CH by the circular-shaped nodes. Therefore, the concept of sub-tree was introduced to ease the process of hole-patching. Sub-trees divide the bigger CHs into smaller sub-holes such that it becomes easier to patch those sub-holes.

4.3 Identification of patch-positions

To identify the patch-positions to be healed by deploying additional SNs, we again identify the center of each subtree based on $E(v)$ as shown in Fig. 8. Now, node 3 is the center for sub-tree 1 and node 8 is the center for sub-tree 2.

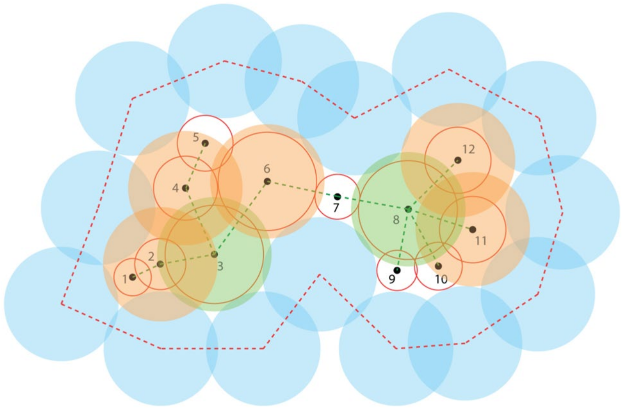


Fig. 9 Hole-patching with extra nodes

Thereafter, one sensor node is placed on the centers of each sub-tree in such a way that the IECs and the patching nodes are concentric. Subsequently, all the 1-D neighbors of each sub-trees are patched in a similar way. Figure 9 shows the final hole-healing where green nodes cover the centers of each sub-tree and red nodes cover the 1-D neighbors of the centers.

The pseudo-code for CHPT is given below:

```
Pseudo-code for CHPT:
Assumptions: Let $R_S$ and $R_C$ be the sensing and communication
  radius and N number of static, homogeneous nodes are being
  deployed in the RoI randomly.
Prerequisites: As per Sect. 3, coverage-hole is detected and hole-
  area estimation is done based on hole-merging.
    – Step 1: List all IECs according to their sizes and join the
      center of two IECs to construct the tree.
    – Step 2: Identify the center of the tree based on the smallest
      eccentricity value.
    – Step 3: The tree is further divided into sub-trees and the
      center of the tree is removed from the list mentioned in step
      1. The center will now act as the Root Node (RN) of the new
      sub-tree to be formed.
    – Step 4: The adjacent 1-D IECs to the centers are removed
      from the list and added to the sub tree.
    – Step 5: If the radius of an IEC is smaller than the correspond-
      ing 1-D IEC, then the IEC is removed from list and added
      to the sub tree. Or if the radius of an IEC is bigger than the
      corresponding 1-D IEC, it acts as the center of another new
      sub-tree.
    – Step 6: Steps 1-4 are repeated for the compilation of the sub-
      tree such that the list of step 1 becomes empty. The sub-tree
      now denotes the location of the hole to be covered.
    – Step 7: Calculate the center of each sub-tree using smallest
      eccentricity value. This center is now the RN for the newly
      formed sub-tree.
    – Step 8: Place one sensor node upon the RN of each sub-tree
      such that IECs and sensor nodes are concentric.
    – Step 9: Repeat step 8 for each 1-D neighbors of each sub-tree
      for optimal hole-healing.
```

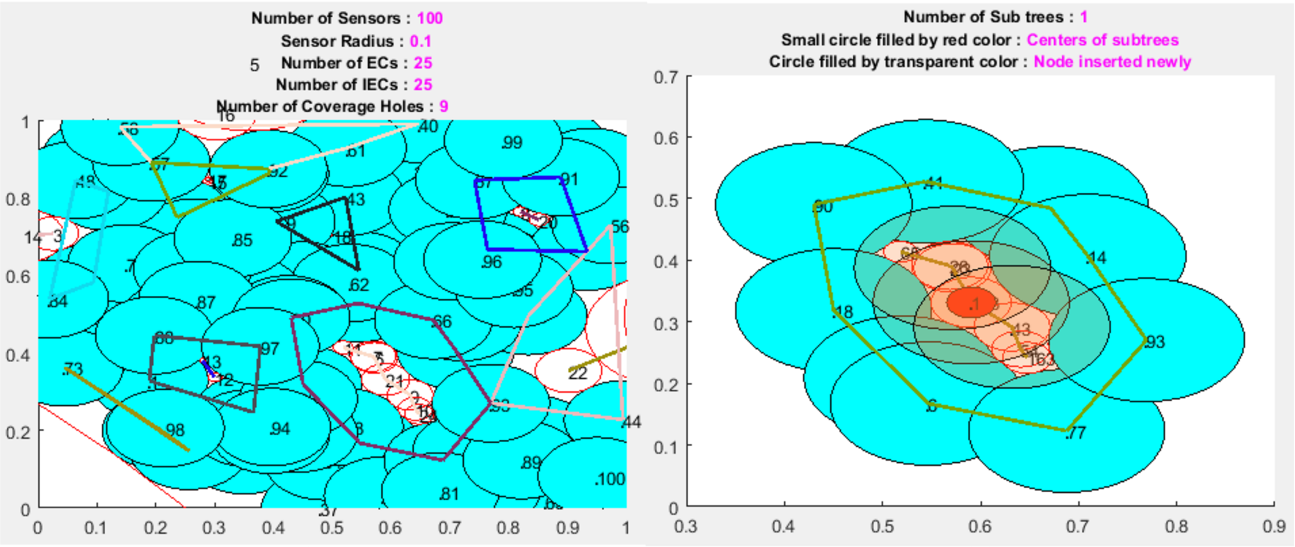
5 Simulation and results analysis

We have simulated the proposed method using MATLAB R2015b. In this proposed work, we have considered the whole scenario in a two-dimensional RoI for simplicity of the calculation. Yet again, we have considered the nodes to be static in nature because the deployment of mobile nodes is currently out of scope of this proposed work. The deployments of the nodes are random in nature for the unbiased hole-creation, so that the set of static nodes can be triangulated and the circumcircle of every triangle can be calculated for identifying the presence of coverage-hole in the RoI. For the minimalism of computation, we have considered a 100 × 100 simulation area where the sensing and communication radius of nodes are regular disks and are equal in size. For experiment purpose, upto 100 static nodes have been deployed randomly assuming R_S varies from 0.10 to 0.15 m. Above parameters used in simulation are displayed in Table 3.

Figure 10a shows the simulation when $N = 100$ and $R_S = 0.10$, the process of hole-detection identifies total 25 no.s of CHs initially, based on empty circle property. Subsequently, the concept of hole-merging based on IEC is applied and the no. of CHs is reduced to 9. Each of these nine holes are then separately represented using trees. Figure 10b displays one out of the nine holes using a tree. The hole-boundary of this particular CH is also displayed in green. This CH is represented using only one sub-tree and finally it is being patched using only three extra nodes. The extra nodes used for hole-patching are assumed to be static in nature and placed to the patch positions according to requirements. The simulation is run for 40 times and the average values are plotted in the graph since the static SNs are deployed randomly, which creates a different deployment every time. From various simulation instances, the relation between number of nodes (N), number of IECs and number of CHs in CHPT is shown in Fig. 11a when sensing radius is fixed, i.e., $R_S = 0.10$.

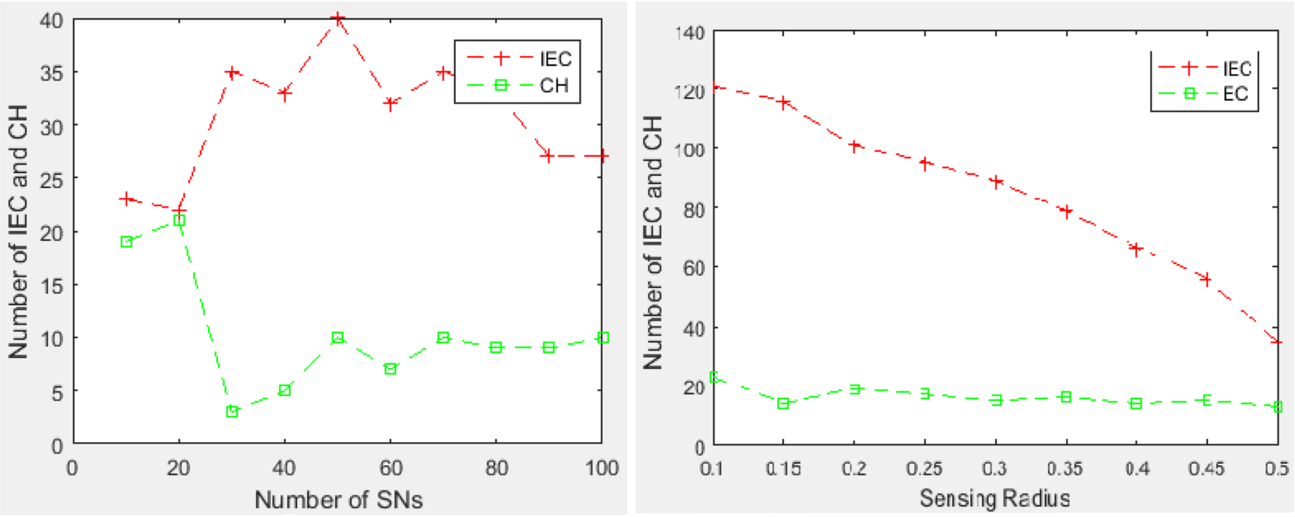
Table 3 Summarization of hole-boundary detection using CG based approach

Parameters	Values
Type of sensors	Static
Deployment nature	Random
Simulation area (input)	100 × 100
Number of nodes (input)	10–100
Rs (input)	0.10–0.15 m
Rc	= Rs
Simulator	MATLAB R2015b



(a) **(b)**

Fig. 10 **a** Tree-formation with $N = 100$ and $R_S = 0.10$; **b** hole-patching with minimum number of nodes = 3



(a) **(b)**

Fig. 11 **a** Relation between N, IEC and CH. **b** Relation between R_S, IEC and CH

It is found that, initially with 20 or lesser number of SNs, the number of IEC and CH was large. This is because, with less nodes, chances of formulation of uncovered region is much higher and this uncovered regions cannot be merged properly due to random placement of less number of nodes. Hence, number of IEC and CH are almost similar. With the increasing number of SNs (by keeping R_S as constant), the number of IEC increases but the number of CHs decreases. The reason behind this fact is, with random deployment of nodes, number of small CHs may increase in various parts of the RoI which may be merged together to reduce the actual number of CHs. When N is constant, if the R_S is increased, the number of IEC as well as CH get decreased. This is because, keeping N constant whenever sensing radius increases, obviously the coverage increases which results lesser CHs. The relation between R_S, IEC and CH is shown in Fig. 11b when N is constant.

To evaluate CHPT, we have compared it with the algorithms—hole-patching using BCP (HPBCP) from Kang et al. (2013) and tree-based hole-healing (THH) from Li and Wu (2016). For comparison between them, we have deployed same number of nodes with a given random arrangement and a similar sensing radius. Then we approached for hole-patching in each method separately. Figure 12a shows the relation between Average percentage of coverage area and no. of extra nodes to be deployed to patch the identified holes inside the network. Here we have discarded the open-holes, i.e., the holes formed in the boundaries. From this Fig. 12a, it is clear that initially up to two extra nodes, CHPT provides average 82.3% coverage, which is slightly better than HPBCP with 80.9% coverage but lesser than THH with 83.8% coverage. The reason is, in CHPT, the initial two extra nodes are placed concentric to only root nodes of the sub-trees and in most of the iterations, at max two sub-trees are formulated from a tree. Therefore, a lot of hole-area remains uncovered in order to patch only the root nodes. But with three or more extra nodes, CHPT shows better performance because CHPT covers the roots as well as the 1-D neighbouring nodes. Since, most of the trees have only two sub-trees, chances of patching maximum area gets improved. The average percentage of coverage increases up to 98.6% with only ten (10) numbers of extra node deployment and is shown in Fig. 12b. With the same extra node deployment, THH and HPBCP provides 95% and 85.8% coverage, respectively.

Again, in Fig. 13, the relation between average no. of extra nodes for full coverage vs. number of nodes in the random deployment has been plotted. When the no. of nodes in random deployment increases, the deployment becomes dense and hence, less no. of coverage-holes are generated. Therefore, the average no. of extra nodes required for full coverage decreases in each of the three methods drastically. For 400 nodes deployment in the RoI, the HPBCP requires 28 extra nodes for full coverage and THH and CHPT need 22 and 16 extra nodes respectively. Among the three methods, HPBCP needs maximum no. of extra nodes, because due to its patching of less intersection points, it can not cover much hole-area. THH method needs moderate no. of extra nodes, since the patching locations are far apart and also the method holds a threshold value of 0.5 m for hole-detection. In CHPT, we also have kept the roots of each sub-tree not as neighboring nodes. Hence, overlapping of coverage-area becomes less and therefore, with less no. of extra nodes, maximum amount of the coverage can be provided in CHPT.

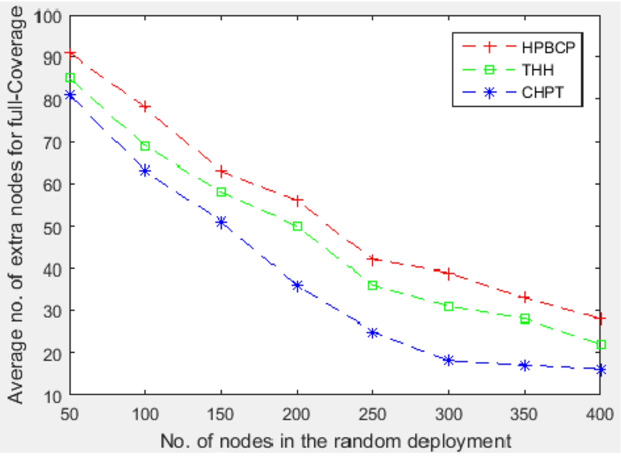


Fig. 13 Improvement in complete coverage with average no. of extra node deployment

Therefore, from the above result analysis and discussion it is found that, CHPT performs better in terms of change in average coverage rate with respect to extra nodes and change in full-coverage with respect to extra nodes deployed in the RoI. Therefore, CHPT reduces the no. of extra node requirement for the patching of the identified holes after successful hole-detection, area estimation and hole-merging.

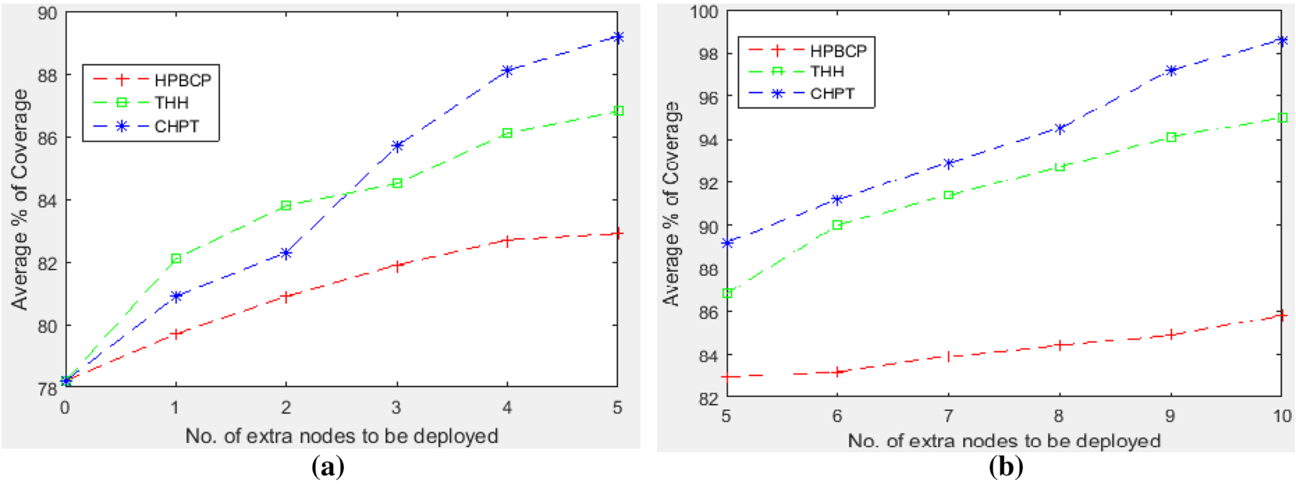


Fig. 12 **a** Improvement in coverage rate with **a** extra five SN deployment **b** extra ten SN deployment

6 Conclusions and future work

In this paper, an improved coverage-hole patching technique is proposed based on the concept of center of tree. As a prerequisite, based on DT and IEC, hole-detection and area-estimation is done. For accurate hole-patching, first the coverage-hole is represented using a tree. Subsequently, the hole-boundary is formulated based on pendant nodes of trees. Thereafter, the different sub-trees are formulated from the main tree to divide the coverage-hole into parts. Finally, the center and its 1-D neighbors are patched using concentric sensor nodes to get optimal hole-patching. Comparison with other methods shows that CHPT is promising in terms of improvement in coverage-rate with the deployment of additional SNs and full-coverage with additional nodes. In future, our target will be to patch the CHs using minimum possible nodes. Also a rigorous study is required to formulate a relation between the number of patching nodes and coverage ratio.

Acknowledgements We would like to thank the Software Engineering and Network Lab of NIT Agartala for carrying out the experimental setup required for this work.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

AbdulGhaffar A, Mostafa SM, Alsaleh A, Sheltami T, Shakshuki EM (2020) Internet of things based multiple disease monitoring and health improvement system. J Ambient Intell Humaniz Comput 11(3):1021–1029

Beghdad R, Lamraoui A (2016) Boundary and holes recognition in wireless sensor networks. J Innov Digit Ecosyst 3(1):1–14

Das S, DebBarma MK (2018) Hole detection in wireless sensor network: a review. Recent findings in intelligent computing techniques. Springer, Berlin, pp 87–96

Das S, Debbarma MK (2019a) Node position estimation for efficient coverage hole-detection in wireless sensor network. Comput Sist 23(1):185

Das S, Debbarma MK (2019b) A survey on coverage problems in wireless sensor network based on monitored region. Advances in data and information sciences. Springer, Berlin, pp 349–359

Das S, Kanti DebBarma M (2018) Computational geometry based coverage hole-detection and hole-area estimation in wireless sensor network. J High Speed Netw 24(4):281–296

Ghaffari A (2019) Hybrid opportunistic and position-based routing protocol in vehicular ad hoc networks. J Ambient Intell Human Comput 20:1–11

Ghosh P, Gao J, Gasparri A, Krishnamachari B (2014) Distributed hole detection algorithms for wireless sensor networks. In: 2014 IEEE 11th international conference on mobile ad hoc and sensor systems, IEEE, pp 257–261

Jing R, Kong L, Kong L (2014) Boundary detection method for large-scale coverage holes in wireless sensor network based on minimum critical threshold constraint. J Sens 20:2014

Kang Z, Yu H, Xiong Q (2013) Detection and recovery of coverage holes in wireless sensor networks. J Netw 8(4):822

Kanno J, Buchart JG, Selmic RR, Phoha V (2009) Detecting coverage holes in wireless sensor networks. In: 2009 17th mediterranean conference on control and automation, IEEE, pp 452–457

Kar K, Banerjee S (2003) Node placement for connected coverage in sensor networks

Khan TF, Kumar DS (2019) Ambient crop field monitoring for improving context based agricultural by mobile sink in WSN. J Ambient Intell Human Comput 20:1–9

Le Nguyen P, Nguyen K, Vu H, Ji Y (2019) Telpac: a time and energy efficient protocol for locating and patching coverage holes in WSNS. J Netw Comput Appl 147:102439

Li W, Wu Y (2016) Tree-based coverage hole detection and healing method in wireless sensor networks. Comput Netw 103:33–43

Li W, Zhang W (2015) Coverage hole and boundary nodes detection in wireless sensor networks. J Netw Comput Appl 48:35–43

Liang J, Liu M, Kui X (2014) A survey of coverage problems in wireless sensor networks. Sens Transducers 163(1):240

Li X, Hunter DK (2008) Distributed coordinate-free hole recovery. In: ICC workshops-2008 IEEE international conference on communications workshops, IEEE, pp 189–194

Liu X (2012) Sensor deployment of wireless sensor networks based on ant colony optimization with three classes of ant transitions. IEEE Commun Lett 16(10):1604–1607

Mellouk A, Senouci M, Aissani A (2013) A localized movement-assisted sensor deployment algorithm for hole detection and healing

Musa A, Gonzalez V, Barragan D (2019) A new strategy to optimize the sensors placement in wireless sensor networks. J Ambient Intell Humaniz Comput 10(4):1389–1399

Ratnasamy S, Karp B, Shenker S, Estrin D, Govindan R, Yin L, Yu F (2003) Data-centric storage in sensornets with GHT, a geographic hash table. Mob Netw Appli 8(4):427–442

Roman R, Zhou J, Lopez J (2006) Applying intrusion detection systems to wireless sensor networks. In: IEEE consumer communications and networking conference (CCNC 2006)

Shen Z, Chang Y, Jiang H, Wang Y, Yan Z (2010) A generic framework for optimal mobile sensor redeployment. IEEE Trans Veh Technol 59(8):4043–4057

Singh P, Chen YC (2019) Sensing coverage hole identification and coverage hole healing methods for wireless sensor networks. Wirel Netw 20:1–17

Wang YC, Tseng YC (2008) Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage. IEEE Trans Parallel Distrib Syst 19(9):1280–1294

Yan L, He Y, Huangfu Z (2020) A fish swarm inspired holes recovery algorithm for wireless sensor networks. Int J Wirel Inf Netw 27(1):89–101

Yang J, Fei Z (2010) Hdar: Hole detection and adaptive geographic routing for ad hoc networks. In: 2010 proceedings of 19th international conference on computer communications and networks, IEEE, pp 1–6

Yan F, Martins P, Decreusefond L (2011) Connectivity-based distributed coverage hole detection in wireless sensor networks. In: 2011 IEEE global telecommunications conference-GLOBECOM 2011, IEEE, pp 1–6

Yu Y, Govindan R, Estrin D (2001) Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks

Zhang G, You S, Ren J, Li D, Wang L (2016) Local coverage optimization strategy based on voronoi for directional sensor networks. Sensors 16(12):2183

Zhang G, Qi C, Zhang W, Ren J, Wang L (2017) Estimation and healing of coverage hole in hybrid sensor networks: a simulation approach. Sustainability 9(10):1733

Zhang Z, Lu Z, Li X, Huang X, Du DZ (2019) Online hole healing for sensor coverage. J Glob Optim 75(4):1111–1131

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.