



Reinforcement Learning-Based Technique to Restore Coverage Holes with Minimal Coverage Overlap in Wireless Sensor Networks

Nilanshi Chauhan¹ · Piyush Rawat¹ · Siddhartha Chauhan¹

Received: 8 November 2021 / Accepted: 28 February 2022 / Published online: 10 May 2022
© King Fahd University of Petroleum & Minerals 2022

Abstract

Coverage holes are the anomalies that can disrupt the coverage and connectivity of a wireless sensor network. It is imperative to equip the sensor nodes with energy-efficient hole detection and restoration mechanism. Existing research works either introduce a new node in the network or use the existing active nodes to recover the coverage loss. The addition of new nodes in the network, after the occurrence of a coverage hole, is not feasible if the area of interest is at a hostile location. The relocation or sensing range customization of active nodes not only results in a constantly changing network topology but also risks the generation of new coverage holes as well as increases the coverage overlapping. Current work presents three algorithms viz., minimal overlapping and zero holes coverage (MO_ZHC), predictable and non-predictable holes recovery scheme (PNP_HRS), and a game theory-based reinforcement learning (GT_RL) algorithm. During the random deployment, the nodes use MO_ZHC to achieve minimal coverage overlapping in the network. After the scheduling round, PNP_HRS utilizes the sleeping nodes to restore the coverage lost, due to the holes. The active nodes are not displaced from their location, but they learn using GT_RL, to select and wake up a sleeping node which can recover the coverage loss in the most energy-efficient manner. The proposed algorithms ensure that the mobility of the nodes is kept minimal for judicious utilization of limited energy resources. The simulation results prove the efficacy of the present approach over the previous research works.

Keywords Coverage holes · Coverage overlapping · Wireless sensor networks · Reinforcement learning · Sensor nodes

1 Introduction

Lightweight, wireless, intelligent sensing devices that can communicate in a distributed manner have made continuous monitoring possible over the past decade. These nodes can be mobile or static and are deployed either deterministically or randomly in the area of interest (AOI) [1]. The prime objective of the deployed sensor nodes is to cover the AOI with the required degree of coverage. The coverage provided by a wireless sensor network (WSN) contributes to its perfor-

mance and serves as a metric to judge the quality of service delivered by it. Many WSN applications, such as battle-field monitoring, real-world habitat monitoring, intelligent warning systems, disaster management, and so on, necessitate constant tracking of the AOI [1]. Manual intervention may not be possible for an AOI situated at an unreachable and harsh location after the node deployment. Therefore, the nodes must be equipped and trained to counter any anomalies affecting the reliability and efficiency of a WSN. Amongst these, coverage holes in the network can result in coverage and connectivity loss which shall be intolerable for applications requiring continuous real-time coverage.

For a randomly deployed AOI, a considerable number of nodes are needed to avoid the occurrence of coverage holes [2]. Holes resulting from random deployment are the first category of coverage holes, usually eliminated during the network initialization to ensure coverage in the AOI. The other two categories are predictable and non-predictable holes. The former involves holes due to complete energy drainage of one or more nodes—the latter involves situations such as damage to nodes, which could be manual or

✉ Nilanshi Chauhan
nilanshi@nith.ac.in

Piyush Rawat
psh.rawat@gmail.com

Siddhartha Chauhan
sid@nith.ac.in

¹ Computer Science and Engineering Department, National Institute of Technology Hamirpur, Hamirpur, Himachal Pradesh 177005, India



natural. Hardware and software failure of a node can also lead to the untimely end of a node. It is vital to avoid the predictable holes and quickly recover from the non-predictable holes to prevent any coverage and connectivity loss. Therefore, maintenance of a WSN after its deployment is as vital as its deployment and initialization. The solution to coverage holes can be either centralized or distributed. A centralized-based approach requires the sink node to detect and restore coverage loss due to the coverage holes. This process can cause a delay in coverage restoration and induce unnecessary communication overhead in the network [3]. The increased data transmissions also increase the probability of collisions in the network. On the other hand, distributed-based solutions to coverage holes are comparatively faster, scalable, and reliable [4]. Therefore, the present study utilizes a distributed approach to resolve the coverage holes.

In the present study, the AOI is randomly deployed with a large number of nodes to prevent the first category of coverage holes. However, this leads to coverage overlap amongst the sensor nodes. Minimal overlap with zero holes coverage algorithm is proposed to minimize the coverage overlapping with the help of sensing range customization and sleep/wake scheduling. MO_ZHC schedules maximal nodes to a sleep state without generating any coverage holes. Predictable and non-predictable holes recovery scheme is proposed to deal with predictable and non-predictable holes. PNP_HRS uses the mobility and sensing range customization to recover coverage loss generated by the coverage holes. Both MO_ZHC and PNP_HRS achieve Nash equilibrium with the help of Game theory-based reinforcement learning (GT_RL). The active set of nodes periodically check for coverage holes, and the sleeping nodes near the coverage hole are used to recover the holes efficiently. The nodes use GT_RL to decide the optimal action of (i) moving a node, (ii) sensing range customization, (iii) combined action of moving and sensing range customization, (iv) neither moving nor customization to restore a coverage hole. The number of sleeping nodes benefits coverage restoration. By waking up only the sleeping nodes, the holes are quickly recovered without disrupting the coverage maintained by the active nodes. It also ensures that no new holes are created during the recovery process, and the minimized percentage of coverage overlap remains the same. The contributions of the present paper are summarized below:

- (i) Recovery of coverage holes without displacing the active nodes from their location to avoid the occurrence of new coverage holes while repairing an existing one. The sleeping nodes are utilized efficiently to recover the coverage in the holes in a distributed manner with the help of minimal mobility and sensing range customization.

- (ii) Early convergence of GT_RL algorithm due to simultaneous learning of nodes located in different cell partitions.
- (iii) The nodes take decisions based on their local view of the network which minimizes the communication overhead.
- (iv) Minimal coverage overlapping is achieved in AOI with the help of proposed MO_ZHC algorithm. Both predictable and non-predictable holes are recovered using the proposed PNP_HRS.

The following section contains a brief overview of the literature on coverage hole repair techniques and a preview of the research gaps. The network model and key terminologies are found in Sect. 3. The proposed methods and their pseudocodes are illustrated in Sect. 4. The findings and discussions are presented in Sect. 5. Finally, in Sect. 6, conclusions are drawn.

2 Literature Review

To recover the coverage loss, coverage hole recovery solutions either use sensor node mobility or configure the sensing range of the nodes. The former category is also called a relocation-based scheme, while the latter is often termed as a power-transmission-based scheme. In both situations, energy consumption and computation overhead should be minimal as the sensor nodes cannot afford to spend too much energy and computational power on maintenance.

In the Hole Repair Algorithm (HORA), the one-hop neighbours of the dead node are aware of the coverage hole generated by it [3]. Mobile nodes recover the hole based on their location and the degree of coverage overlapping. The holes are recovered in a distributed manner using a game theory-based approach [4]. Both node relocation and sensing range customization techniques are utilized to cover the holes with minimum coverage overlap. This technique does not restrain the distance of relocation for a node, resulting in energy drainage of mobile nodes. Also, each node chosen to cover the hole takes a combined action of relocation and range customization, which can be avoided in some instances where both actions are not required together.

The need to minimize coverage overlapping while restoring the coverage loss created in the holes is given prime consideration [5]. An algorithm called the Healing Algorithm of Coverage Hole (HACH) is implemented to achieve full coverage without coverage holes and minimal overlap. HACH relocates minimal nodes to the best positions to repair the hole. The intersection points of overlapped sensing ranges of the nodes are used to detect the coverage holes and their respective locations. Relocation of nodes does not prove valuable when the hole is small and redundant nodes are not available.

Each node detects the presence of holes by exchanging information with its neighbours in the Coverage Hole Detection and Restoration algorithm (CHD-CR) [6]. During the hole restoration phase the node with higher residual energy and least distance from the hole restores coverage by increasing its sensing range to maximum limit. Increasing the sensing range to maximum limit saves time for estimating the required sensing range to cover the hole but increases the chances of coverage overlap.

Cascaded neighbour intervention algorithm chooses the nodes based on the residual energy, coverage redundancy, and relocation distance to repair the coverage holes [7]. This process might lead to the creation of new holes in addition to the existing ones. These new holes are simultaneously resolved using cascaded neighbour intervention repeatedly. A significant drawback is the creation of new holes while recovering an old one. The possibility of abrupt failure of a sensor node is not considered.

A greedy approach-based hole detection and restoration algorithm are provided in [8]. The presence of holes is detected using the intersection points-based method. After a node detects the hole, it sends the information to the sink node. New nodes are introduced in the network to recover the holes. Sending the information to the sink generates unnecessary overhead. It also means that the approach is not entirely decentralized.

Two distributed algorithms, viz., the Hole boundary discovery procedure and Hole recovery algorithm, are employed to detect and restore the coverage holes automatically [9]. The hole detection algorithm uses the concept of perimeter coverage and intersection points. Sensing range customization is not utilized for heterogeneous sensor nodes, resulting in the overhead of choosing a redundant node with the required radius.

The nodes learn by choosing the action of changing the sensing radius and relocating themselves with the help of reinforcement learning [10]. Relocation is limited to one hop. A standard intersection method is applied to detect the holes [11]. The node which detects a hole relocates itself to restore the coverage only if it does not generate any new holes; otherwise, it chooses a new source node randomly. Moving a random node to the hole location could increase the energy expenditure and the possibility of coverage overlapping.

A similar procedure is followed in [12], where the intersection method detects the holes. But during the hole recovery phase, a new node is placed at the centre of the coverage hole. Although adding a new node bypasses selecting a suitable node, it does so at the expense of adding new nodes to the network when redundant nodes can perform the same function.

A distributed reinforcement learning-based protocol called coverage and connectivity maintenance (CCM-RL) is given in [13]. It utilizes the sensing range customization to

activate a minimum number of nodes for coverage and connectivity in the target area. Although CCM-RL minimizes the coverage overlapping but it neglects the occurrence of coverage holes during this process.

A fully distributed algorithm is suggested to detect and repair coverage holes [14]. First, the algorithm detects a failed node then identifies the size and position of the hole. Even though the minimum movement of nodes is incorporated, several nodes need to be displaced to restore a coverage void.

A novel distributed self-healing algorithm called distributed hole detection and repair (DHDR) is presented [15]. DHDR handles both hole detection and repair with the help of already deployed nodes. The nodes share information and coordinate their movements to relocate in such a way that the coverage hole is restored without disrupting the existing coverage and connectivity. All the nodes bounding the coverage holes are candidates for hole repair. Movement of nodes can be further reduced by allowing the nodes to customize their sensing radius. Mobile sensor nodes are dispatched to heal the coverage holes detected by a set of static sensor nodes [16]. The authors have proposed an improved self-organized mapping (SOM) for coverage hole restoration. The nodes for coverage hole restoration are chosen using a fuzzy system which assigns eligibility to each mobile sensor node. Eligibility of a sensor node depends on its moving distance and residual energy. SOM detects the coverage holes after their occurrence.

The occurrence of coverage holes due to unoptimal deployment of sensor nodes [17]. The authors have combined the gradient algorithm-based recovery method with clustering technique to detect redundant sensor nodes. These nodes are relocated to restore the coverage holes, but the movements of the sensor nodes have not been minimized. An intersection-based method has been utilized to detect the boundary of coverage holes [18]. But during the hole recovery phase, a new node is placed at the centre of the coverage hole. Although adding a new node bypasses the process to select a suitable node, but at the expense of adding new nodes to the network when redundant nodes can perform the same function.

It has been observed from the literature review that solutions to the problems of coverage overlapping and coverage holes go hand in hand. While the former happens due to unnecessarily active sensor nodes in one area, the latter occurs because of the deficiency of active sensor nodes. Therefore, a solution to either of these can negatively impact the problem if both are not acknowledged together.

A solution to restore the coverage in the holes involves sensing range customization and relocation. Without proper trade-offs, both solutions can increase the coverage overlapping. Therefore, during the initialization of the network, minimal nodes are activated with zero coverage holes and minimal overlap in the present study. After the initialization,

Table 1 Categorization of coverage optimization algorithms based on important parameters

Algorithm	Hole recovery method		Existence of holes						
	N-node	R-node	DRD	ARD	M-Nodes	SRC	COM	HR	Learning
HORA [3]	×	✓	×	✓	✓	×	✓	✓	×
Hybrid CH recovery [4]	×	✓	×	✓	✓	✓	✓	✓	✓
HACH [5]	✓	✓	✓	×	✓	×	✓	✓	×
CHD-CR [6]	×	×	×	✓	×	✓	×	✓	×
CNI [7]	×	✓	×	✓	✓	×	×	✓	×
Greedy [8]	✓	×	×	✓	×	×	×	✓	✓
DCHD [9]	×	✓	✓	✓	✓	×	×	✓	×
Hybrid recovery [10]	×	✓	×	✓	✓	✓	×	✓	✓
Coverage enhancement [11]	×	✓	×	✓	✓	×	×	✓	×
CBHC [12]	✓	×	×	✓	✓	×	✓	✓	×
CCM-RL [13]	×	×	✓	×	×	✓	✓	×	✓
DHDR [15]	×	✓	×	✓	✓	×	×	✓	×
SOM [16]	×	✓	×	✓	✓	×	×	✓	✓
Proposed approach	×	✓	×	✓	✓	✓	✓	✓	✓

The parameters namely, N-node, R-node, DRD, ARD, M-nodes, SRC, COM and HR represent new node, redundant node, during random deployment, after random deployment, mobile nodes, sensing range customization, coverage overlap minimization and hole recovery, respectively

the active set of nodes periodically checks for the occurrence of coverage holes. A few existing algorithms related to coverage optimization are categorized in terms of important parameters as shown in Table 1. The classification parameters used are hole recovery method (new node i.e. N-node is introduced to recover the hole or existing redundant node i.e. R-node is used), existence of holes (holes exist during the random deployment (DRD) or after the random deployment (ARD)), inclusion of mobile nodes in the network (M-node), sensing range customization (SRC), coverage overlap minimization (COM), inclusion of hole recovery method (HR) and the use to learning to recover the holes. The hole recovery algorithm proposed in the present study, works energy efficiently to recover the coverage without generating any new holes and increasing the existing minimal coverage overlapping.

3 Important Terminologies and Network Model Assumptions

Some important terminologies used in this study are defined as follows:

- *init_node*: This is the node which initiates the learning process inside a cell during the starting of a scheduling round.
- *detector_node*: It is the node responsible for detecting a coverage hole.
- *rescue_node*: Node chosen by the *detector_node* to recover the hole.

- Neighbour nodes: Nodes within the transmission distance of a node and located inside cell boundary. These are denoted by the set $\{NN\}$.
- First neighbours: Nodes within the sensing distance of a node and located inside the cell boundary. These are denoted by the set $\{FN\}$, such that $\{FN\} \subset \{NN\}$
- Single hop distance: It is equivalent to the sensor node's transmission range.
- First neighbour coverage (*FNC*): Sensing the region of a node that is surrounded by its first neighbours.
- Intersection point (*IP*): The points of intersection are formed by the overlapped sensing areas of two or more sensor nodes.
- Boundary intersection point (*BIP*): The points where the sensing field of a sensor node located inside a cell, intersects with the cell boundary.
- B-node: The node closer to the boundary of the cell as compared to its centre.

The sensor nodes are randomly deployed in the AOI. The following are the system model and the assumptions considered:

- The network is homogeneous at the beginning. Each node can customize its sensing radius within a defined range and move over short distances. The network becomes heterogeneous when nodes set their sensing range individually based on their requirements and learning.
- The coverage holes can occur due to both sudden node failure and battery drainage.

- (iii) A node localization algorithm [19] informs each node of its position. The node also has details about its neighbours' location, state, and residual energy.
- (iv) Basic energy dissipation for transmission and reception of data packets is adapted from the work based on energy-efficient coverage [20,21].
- (v) Transmission radius (R_c) of each node is equal to twice the maximum possible value of the sensing radius (R_s).
- (vi) The AOI is divided into multiple square cells with each side of the cell equal to the sensing radius of the nodes during random deployment. Hence for the AOI of size $L \times L$, number of cells required for partitioning is:

$$| Cells | = \frac{L \times L}{R_s \times R_s} \tag{1}$$

Figure 1 shows the cell partitioning of a randomly deployed AOI.

- (vii) For N number of nodes deployed in the AOI, each node S_j , resides in a $cell_i \in Cells$, where $i \leq | Cells |$ and $j \leq N$. It is defined formally using a tuple n_j constituting of seven attributes. $n_j = \{id_j, l_j, R_c[j], R_s[j], e_r[j], NN^j, FN^j\}$ where id_j is a unique identity number of S_j , and l_j is the location of S_j . $R_c[j]$ represents the communication range and $R_s[j]$ is the sensing range of S_j which is customizable. $e_r[j]$ is the residual energy, NN^j and FN^j are the sets of neighbour nodes and first neighbours of S_j , respectively. NN and FN contain 3-attribute tuples pertaining to the neighbour nodes and first neighbours of S_j . A tuple belonging to k th neighbour and m th first neighbour can be accessed as NN_k^j and FN_m^j , respectively, for $k \leq | NN^j |$ and $m \leq | FN^j |$.
- (viii) Due to the wider scope of random deployment, the nodes are deployed randomly in the AOI. In the simplest form of random deployment (simple diffusion), the nodes are scattered from above the ground and the randomly generated node locations are defined by a probability density function (pdf) as follows [22]:

$$pdf(X) = \frac{1}{2\pi\sigma^2} H(\| X - C \|), H(v) \triangleq e^{-\frac{v^2}{2\sigma^2}}, X \in \mathbb{R} \tag{2}$$

where X defines the location of a node, C refers to the ground reference point above which lies the scattering point. σ^2 refers to the variance of the random distribution of the nodes.

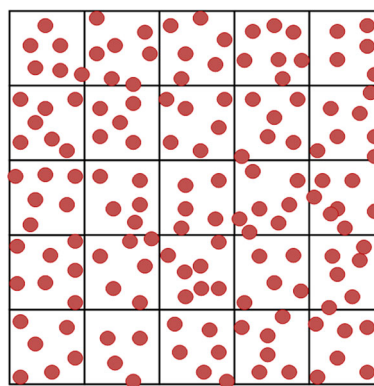


Fig. 1 Square cell-based partitioning of the randomly deployed WSN

4 Proposed Methodology

In this section, three algorithms viz., MO_ZHC, PNP_HRS, and GT_RL are proposed to achieve an energy-efficient coverage scheme with augmented coverage hole restoration and minimal overlapping. The low cost and small size of sensor nodes allow the scattering of a considerable number of sensor nodes in the AOI, which minimizes the probability of the occurrence of coverage holes due to random deployment. But the more the nodes scattered during random deployment, the more is the overlapping in the AOI. Such a situation results in redundant sensing, which increases the redundancy of the transmissions of data packets towards the sink. Too many redundant communications increase the chances of collisions in the network and lead to unnecessary energy drainage of redundant sensor nodes. MO_ZHC puts a maximum number of redundant nodes to sleep in a scheduling cycle, minimizing the coverage overlapping, resulting from random deployment of substantial sensor nodes.

The detector nodes use PNP_HRS to recover the coverage holes occurring during a scheduling round. The dying nodes or nodes facing abrupt hardware/software failures could be responsible for the generation of these holes in the network. Every active node periodically checks for the occurrence of non-predictable holes. The detector node takes energy-efficient decisions involving relocation and range customization to recover the holes while maintaining the percentage of coverage overlap, earlier minimized using MO_ZHC. In the case of predictable holes, when a node reaches the residual energy level lower than the threshold energy (e_t), it informs the initiator node so that a coverage hole due to its death can be prevented.

MO_ZHC and PNP_HRS utilize game theory-based reinforcement learning to achieve Nash equilibrium in the WSN environment. Learning is initiated at the initiator node in each cell concurrently. In [13], the sensor nodes were chosen randomly to run the learning algorithm sequentially. This process can result in a delay in providing or restoring cov-

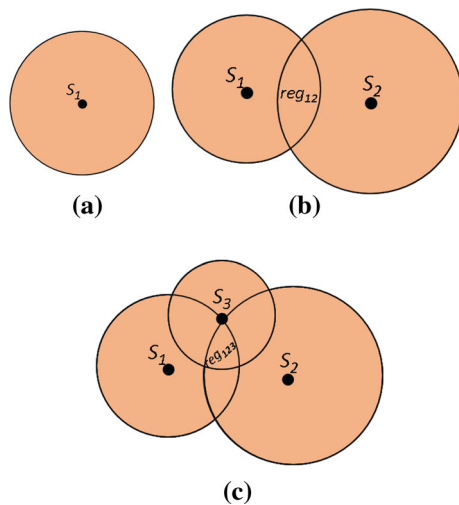


Fig. 2 **a** Coverage overlapping with degree 0; **b** coverage overlapping with degree 1; **c** coverage overlapping with degree 2

erage in the AOI. Therefore, in the present study, initiator nodes are chosen from each cell, which simultaneously executes the learning process. In the proposed algorithms, nodes belonging to different cells do not interfere with each other's decisions. Thereby allowing the simultaneous execution of learning algorithms by the nodes in different cells possible.

4.1 Minimal Overlapping and Zero Holes Coverage Algorithm (MO_ZHC)

Coverage overlapping results from the overlapped sensing areas of sensor nodes. Degree of coverage overlapping refers to the number of times a particular region (*Reg*) is covered by overlapped coverage. The degree of overlap is 0 if only a single node covers *Reg*, the degree is equal to 1 if the sensing ranges of two nodes overlap the region. Similarly, the degree is equivalent to 2 if the sensing ranges of three nodes overlap the *Reg*. Figure 2 shows the scenarios for up to 2 degrees of coverage overlapping. It is important to note that the degree of coverage overlapping is not the same as the degree of coverage. A degree of coverage overlapping equal to 0 results in a single degree of coverage. Likewise, degree 1 and degree 2 coverage overlapping results in double and triple degrees of coverage, respectively. A region *Reg* in the AOI can be divided into various regions based on the degrees of coverage overlapping experienced by them. The AOI is said to have complete coverage when all the cells have each point covered by at least one node. Cell coverage can be estimated by Eq. (3).

$$\eta = \frac{[Ar(cell_i) \cap \{Ar(S_1) \cup Ar(S_2) \cup Ar(S_3) \cup \dots \cup Ar(S_{N_c})\}]}{Ar(cell_i)} \quad (3)$$

where N_c is the total number of nodes inside $cell_i$, $Ar(cell_i)$, and $Ar(S_j)$ for $j \leq |N_c|$, represent the area of the i th cell and the sensing area of the node S_j , respectively. A cell is fully covered when $\eta = 1$. MO_ZHC aims to keep coverage overlap to a minimum while avoiding coverage holes in the network. As a result, minimal number of nodes are involved

in a single scheduling round to maintain coverage for the AOI.

Algorithm 1 details the steps for achieving minimum overlapping and zero holes coverage. After the random deployment, the AOI is partitioned into cells of size $R_s \times R_s$. This is done so that even a single node inside the cell could cover at least 80% of the cell area. Total number of cells in the AOI can be determined using eq(1). Each cell is defined using four attributes, viz., the number given to the cell (*cellno*), the coordinates of intersection of the diagonals and the maximum and minimum values of the x and y coordinates of the points inside a cell. $cell_i = \{cellno, (x_c, y_c), (x_{max}, y_{max}), (x_{min}, y_{min})\}$. For each $cell_i \in Cells$, the sink node determines a node S_j which is located at the least distance from (x_c, y_c) and sends the cell tuple to that node. Using these values each node can identify the cell number it belongs to. S_j broadcasts this message to the nodes within its transmission distance. Using this tuple each node in the AOI identifies the cell it belongs to. A node might receive the cell tuple from a number of nodes if it is located near the boundary of a cell. Hence, the node sends back an acknowledgement message only to the node with which it shares a cell. Each node also identifies its neighbour nodes (NN) and first neighbours (FN) within the same cell, based on their distance from it. Each node in the AOI estimates the first neighbour coverage percentage (FNC_j) of its sensing area. It is computed as follows:

$$FNC_j = \frac{\sum_{m=1}^{|FN^j|} (Ar(S_m) \cap Ar(S_j))}{Ar(S_j)} \times 100 \quad (4)$$

where $(Ar(S_m) \cap Ar(S_j))$ denotes the area of S_j that is distinctly occupied by $S_m \in |FN^j|$. Each node S_j creates a first neighbour coverage list (FNC_List) and initializes it with its own *id* and the corresponding FNC_j value. S_j broadcasts its own FNC_j value to all the neighbour nodes within the cell. S_j updates its own FNC_List by appending the FNC percentage values received from its neighbours. At the end of FNC_List updation process, all the nodes belonging to a particular cell are aware about the FNC values corresponding to their neighbour nodes. The list is then sorted in the ascending order of the FNC values. Each node calculates a timer $t[j]$ which depends on the FNC value and the time taken by a node to make a learning decision i.e. Δt . The value of Δt is very small and is estimated as 0.1725. This value is 20% higher than the actual average time of state transition to ensure that two nodes in the same cell do not take an action at the same time. If two or more nodes in a cell start to make decisions about their actions together, it could result in conflicts.

The node with the least FNC value has a timer value of 0 and is called the initiator node (*init_node*). There is one

Algorithm 1: Minimum Overlapping and Zero Holes Coverage Algorithm

```

Input:  $cell_i, n_j, R_s, e_r[j], l_j$ 
Output:  $FNC\_List, FNC_j, \Delta rs$ 
1 Randomly deploy  $N$  nodes in the AOI of size  $L \times L$ .
2 Divide the AOI into cells of size  $R_s \times R_s$ ,  $R_s \leftarrow$  sensing range of the nodes during deployment.
3 Each  $cell_i = \{cellno, (x_c, y_c), (x_{max}, y_{max}), (x_{min}, y_{min})\}$ ,  $cell_i \in Cells$ .
4  $FNC\_List = \{\}$ 
5 foreach node  $S_j \in cell_i$  do
6   Estimate  $n_j = \{id_j, l_j, R_c[j], R_s[j], e_r[j], NN^j, FN^j\}$ .
7    $NN_k^j = \{id_k, l_k, e_r[k]\}$ , for  $k$ th neighbour node,  $k \leq |NN^j|$ .
8    $FN_m^j = \{id_m, l_m, e_r[m]\}$ , for  $m$ th first neighbour,  $m \leq |FN^j|$ .
9   Estimate the percentage of first neighbour coverage in the sensing area of  $S_j$  as  $FNC_j$ .
10  Broadcast the  $FNC_j$  value to the neighbour nodes.
11  Initialize the  $FNC\_List$  by appending own  $FNC$  value to the list.
12   $FNC\_List = \{(S_j, FNC_j), append\}$ 
13  Complete the  $FNC\_List$  by appending the  $FNC$  values received from the first neighbours and other neighbour nodes.
14   $FNC\_List = \{(id, FNC), append\}$ , such that  $id \in FN^j$  OR  $id \in NN^j$ .
15  Sort the  $FNC\_List$  in the ascending order of  $FNC$  value.
16   $FNC\_List = sort(FNC\_List(2), ascend)$ 
17  while  $FNC\_List \neq Null$  do
18     $pos_j = index((S_j, FNC_j), FNC\_List)$ 
19     $t[j] = (pos_j - 1) \times \Delta t$ ,  $\Delta t$  is the estimated time of a node to decide its state and action.
20     $init\_node =$  node with least  $t[j]$ .
21 foreach node  $S_j$  do
22   if  $t[j] \leftarrow 0$  then
23     if  $FNC_j == 100$  then
24       Action ( $S_j$ )  $\leftarrow$  sleep.
25     else
26       Identify the neighbours with  $id \in FN^j$  and the least area of intersection.
27       Compute the intersection points and store them in the set  $IP[j]$ .
28       if  $S_j \in B - nodes$  then
29         Identify the cell boundary intersection points and store them in  $BIP[j]$ .
30       Estimate  $\Delta rs$ , such that the intersection points  $(x_p, y_p) \in IP[j]$  or  $BIP[j]$  remain covered,  $p$  is the total number of intersection points of  $S_j$  including  $IPs$  and  $BIPs$ .
31       if  $\Delta rs > 0$  then
32         Compute the new range as  $R_s[j] = R_s[j] - \Delta rs$ .
33         Action( $S_j$ )  $\leftarrow$  Customize the range.
34         Action( $S_j$ )  $\leftarrow$  maintain active state.
35       else
36         Action( $S_j$ )  $\leftarrow$  maintain active state.

```

initiator node in each cell. The nodes in different cells can start the learning process together. The nodes in the same cell perform the learning process one after another as their timer expires to avoid any conflicts. Each node identifies its FNC state i.e. full first neighbour coverage or partial first neighbour coverage. Full FNC implies that the sensing area of a node is completely covered by its first neighbours. In this case, the node goes to sleep mode and broadcasts its new state to its neighbours within the cell. On the other hand, if the node is in the state of partial FNC then it estimates Δrs which denotes the sensing range customization variable. A node identifies the set of intersection points (IP) with the first neighbours which contribute the least to its FNC value. In case a node belongs to the set of B-nodes, it finds the intersection points with the cell boundary (BIP). Final estimated value of Δrs must be such that all the intersection points belonging to the set IP and BIP must remain covered after sensing range customization. If a positive value to this variable can be found, then the sensing range is decreased by Δrs and the node remains active. But if a positive value cannot be found then no action is taken by the sensor node.

4.2 Predictable and Non-predictable Holes Recovery Scheme (PNP_HRS)

When the algorithm (MO_ZHC) converges, the AOI has minimal coverage overlapping with minimal active nodes and zero coverage holes. The second algorithm, PNP_HRS, is executed by the active set of nodes to prevent the occurrence of the predictable holes and restore the coverage loss caused by non-predictable coverage holes in the shortest possible time span. Algorithm 2 provides steps for restoring the predictable and non-predictable coverage holes. The active nodes in the AOI periodically check for the occurrence of non-predictable coverage holes. The hole detection is done using the intersection points-based algorithm as discussed in [14]. The nodes learn to make optimal decisions for restoring the holes using reinforcement learning.

Each node checks for the occurrence of coverage holes when the timer, t_h expires. The value of t_h depends on the total expected lifetime of the WSN and is estimated in step 3. Δx extracts a time period from T_E and generates a small period of time after which hole occurrence is checked. The random variable, $nrand_j$ is used to randomize the value of t_h . The sensor node which detects the coverage is designated as the *detector_node*. The *detector_node* is responsible for determining the hole location (x_H, y_H) and the hole boundary using the intersection points-based algorithm, discussed in [14]. The boundary identification involves the determination of uncovered intersection points ($UIPs$). In Fig. 3, I_1, I_2, I_3 are covered intersection points ($CIPs$) which are enclosed by the sensing areas of S_5, S_4 and S_6 , respectively. However, points I_4, I_5, I_6, I_7 and I_8 are the $UIPs$ as they

Algorithm 2: Predictable and Non-predictable Hole Recovery Scheme

Input: T_E, t_h, e_r, e_t
Output: $(x_h, y_h), action_list$

```

1 Part(a) Non-predictable coverage holes recovery
2  $T_E \leftarrow$  Total expected lifetime of the WSN.
3  $t_h \leftarrow T_E \times \Delta x \times nrand_j, \Delta x = 0.0125, nrand_j \in (0, 1]$ .
4 foreach node  $S_j$  in the  $i$ th cell do
5   Check for the occurrence of coverage hole after  $t_h$  expires.
6   if hole  $H$  is found then
7      $S_j \leftarrow detector\_node$ 
8     Generate the hole location  $(x_H, y_H)$ .
9     Find the rescue_node.
10    Send a wake_up_signal to the rescue_node.
11    Wait to receive ACK_signal and send the coordinates of
    the hole  $H$  with appropriate action_list.
12    Rescue_node follows the action_list.
13    Estimate the cost of hole recovery.
14 Part(b) Predictable coverage holes recovery
15 foreach node  $S_k$  in the cell $_i$  do
16   if  $e_r < e_t$  then
17     Update init_node about the potential hole.
18     Send HELP  $\langle POT\_HOLE, l_k \rangle$ .
19   if HELP message is received then
20     Goto Step 7
  
```

are not covered by the sensing range of any node, as seen in Fig. 3. Hence, they form the boundary of hole H . Out of these *UIPs*, two points with maximum distance (d_{max}) are determined. The point of bisection of straight line connecting these two points is the location (x_H, y_H) of the hole H as well as the new location of the *rescue_node*. The new required sensing radius of the *rescue_node* to cover H is equal to $\frac{d_{max}}{2}$. The *detector_node* is responsible for selecting a *rescue_node*. The *detector_node* identifies a candidate set of nodes and chooses the *rescue_node* based on the hole restoration time and the energy consumed in the process. If the estimated hole restoration time exceeds the average expected hole restoration time for a low energy consuming *rescue_node* option then the *rescue_node* has to be chosen by trading off energy for time.

In case of predictable holes, coverage restoration mechanism is the same as discussed for non-predictable holes recovery. A node S_k with low energy levels i.e. $e_r < e_t$ informs the *init_node* about the potential coverage hole by sending a help message as *HELP* $\langle POT_HOLE, l_k \rangle$. l_k denotes the location of S_k . The hole location x_H, y_H is the same as l_k . The threshold value of residual energy is considered as 15% of the initial energy value. The value is taken such that the node can inform the initiator node about the potential coverage hole and the rescue node could be found and activated with necessary actions regarding relocation and/or sensing range customization. The hole boundary comprises the sensing area periphery of S_k . *init_node* acts as

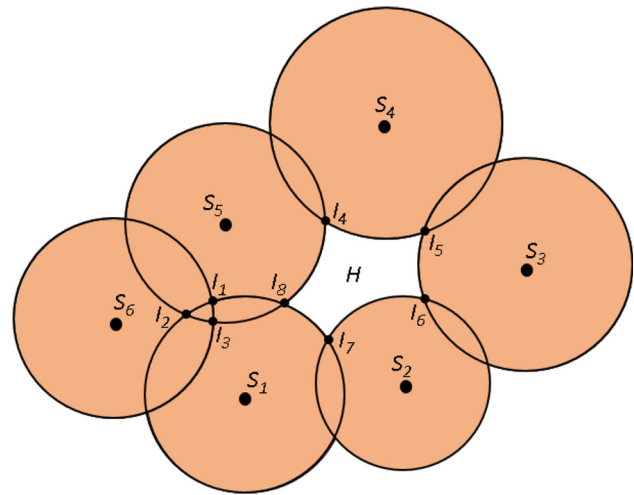


Fig. 3 Intersection points marking the boundary of coverage hole H

the *detector_node* and starts looking for a *rescue_node* in the vicinity of x_H, y_H . The *rescue_node* is always searched within one hop distance from the coverage hole location so that the displacement is never more than a single hop. This ensures minimum mobility which is very crucial for an energy constrained WSN. It is vital to note that a dying node does not look for a *rescue_node* itself because due to its low energy levels, it will not be able to choose an optimal candidate. Therefore, this responsibility is delegated to the *init_node*.

Following candidate set of nodes is identified by the *detector_node* to recover both predictable and non-predictable holes.

- node with current location at the least distance from (x_H, y_H) and which does not need range customization.
- node which doesn't need to move to the new location and can cover the hole from its current location by sensing radius customization.
- node which needs to be moved to the new location along with sensing radius customization.
- node which can cover the hole without any movement and without needing any radius customization.

Point (d) above, refers to a scenario when the candidate node is very close to the hole location and its sensing radius is already larger than the radius of the coverage hole. Based on these candidates, *detector_node* chooses the *rescue_node* and decides the *action_list*.

4.3 Game Theory-Based Reinforcement Learning (GT_RL) Using MO_ZHC and PNP_HRS

Reinforcement learning-based AI and game theory are two integral parts of the multi-agent reinforcement learning.

Game theory is a mathematical framework used to describe and analyse the behaviour of multiple agents which coordinate to achieve a global optimum [23,24]. Each agent has an individual goal to mark its individual performance and a global goal to specify the performance of the environment it interacts with. The results of a decision made by an agent affects the decisions of others. Either all the agents work collaboratively to achieve a single goal, or each agent can have their own local as well as global goals. A game is formally described as $\mathcal{G} = \{\lambda, A, \mu\}$. These attributes λ, A, μ represent three fundamental components of the game, where,

- (i) λ represents the set of players or the learning agents. $\lambda = \{1, 2, \dots, n\}$, where n is the total number of players. All the players except the i th player are represented collectively by $-i$.
- (ii) A represents the set of strategies/actions belonging to each player. $A = \{A_1, A_2, \dots, A_n\}$ where A_i is the strategy opted by i th player which is basically a set of actions taken by it.
- (iii) Each player has a reward function which maps its actions to the positive reward value or the expected payoff. μ is a reward function for each player. $\mu_i : A \rightarrow \mathbb{R}$.

In the present study, Q -learning is applied to the nodes located in different cells constituting the AOI. Q -learning is a reinforcement learning technique. The nodes learn by engaging with the environment repeatedly. The learning nodes perform an action that alters the state of the system in exchange for a reward. This reward-related knowledge acquired by the agent aids it in making better decisions in the future. ϵ -greedy strategy is used by the nodes to select their actions. An action made by one node has an impact on the actions of other nodes. Q -learning in a multi-agent system like WSN, incorporates the concept of game theory, where the reward is provided to an agent depending on the joint actions of other agents as well. A learning agent uses Q -values to learn optimum methods when attempting to optimize the total reward value. The agent begins with a collection of arbitrary initial values for the variable Q . The Q -values at any point in time (t) are cached in a matrix and updated as follows [25].

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (5)$$

$s \in S$, S is a set of states, $a \in A$ and r is the reward while transiting from state s to s_{t+1} . The reward is estimated on the basis of the effect of a learning agent’s actions on the performance parameters of the environment. α is learning rate ($0 < \alpha \leq 1$) and γ is the discount factor ($0 < \gamma \leq 1$). The degree to which new knowledge supersedes old information is defined by the learning rate. At time t , Q -value is initialized as $Q(s, a)$. s' is the new state at time $t+1$ after action a

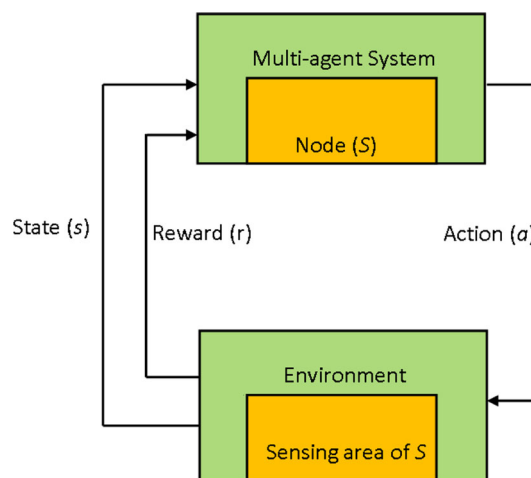


Fig. 4 Reinforcement learning in a multi-agent system

is taken in state s at time t . $[\alpha \gamma \max_{a'} Q(s', a')]$ represents the maximum estimated reward from state s' .

Figure 4 shows the scenario of reinforcement learning in a multi-agent system.

Variable Q in Q -learning represents quality of an action. The more rewarding an action is in a particular state, the higher is the Q -value. Hence each player greedily chooses an action based on the Q -value. ϵ -greedy is used to choose actions based on the Q -value [13]. ϵ -greedy switches between exploration and exploitation. ϵ denotes the probability of exploring the options during a game and $(1 - \epsilon)$ denotes the probability of exploitation. At $\epsilon=1$, maximum exploration happens. In a long run this could lead to a wastage of resources when unrewarding actions are explored. At $\epsilon=0$, maximum exploitation happens. If only exploitation happens then an agent might miss out on some better rewarding actions. Therefore, value of ϵ must be changed regularly to balance between exploration and exploitation of the environment. In the present study, the rate of exploration (ROE) is taken in the interval $[0, 0.55]$. Based on the value of ROE, ϵ is optimized so that a player chooses most rewarding actions for a particular state. There must be further exploration of the environment before the players hit Nash equilibrium. The learning agents will achieve the global optimum faster by using Nash Equilibrium. In this state, each agent has the best strategy in response to strategies expected from other agents. Considering, \mathcal{G} a stochastic game then the strategy (a'_i, a'_{-i}) is attributed as Nash equilibrium state if and only if $\forall a_i \in A_i$ and $\forall a_{-i} \in A_{-i}, \mu(a'_i, a'_{-i}) \geq \mu(a_i, a'_{-i}), \forall i \in \lambda$. The problem of achieving optimized coverage, with minimal overlapping and zero coverage holes, is formulated as a multi-player game. The sensor nodes are the learning agents or players that communicate with one another to reduce coverage redundancy while avoiding network holes. GT_RL illustrates the steps for achieving global optimum. When the

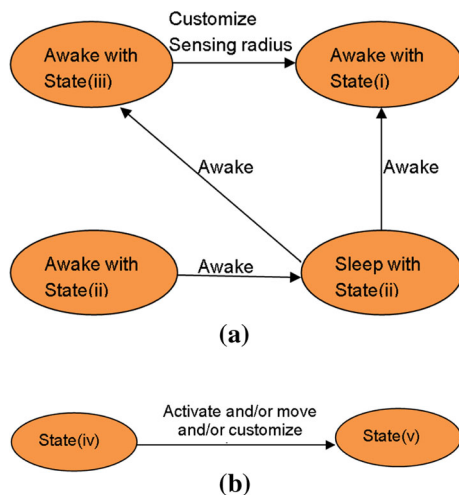


Fig. 5 Possible state transition during **a** coverage overlap minimization; **b** coverage hole recovery

learning process is initiated and the nodes are in exploitation phase, instead of taking random actions, they utilize the MO_ZHC and PNP_HRS algorithms to build the knowledge base. As the nodes enter in the exploration phase, they choose actions with the highest Q -value. A learning agent identifies one of the five states it belongs to with the help of MO_ZHC and PNP_HRS. An action is performed from the action profile. After the action is taken the local and global states are affected with respect to the parameters such as $AWCOP$, coverage rate, energy consumption and hole restoration time, reward is generated, and Q -values are updated. As the agent learns, it uses ϵ -greedy to choose an action which corresponds to highest Q -value. The agents strive to achieve Nash-equilibrium. When the agents are in Nash equilibrium, each agent has the best response action with respect to other agents in the game. An action profile for the agents does not belong to Nash equilibrium until there is a possibility for at least one agent to improve the payoff but changing the action in the action profile. A learning agent node can be in one of the following states during its lifetime:

- (i) Partial FNC with minimal coverage overlapping: In this state, the coverage overlapping cannot be reduced further without generating a coverage hole.
- (ii) Complete FNC with the first neighbours providing full coverage to the sensing area of the node.
- (iii) Partial FNC with coverage overlapping which can be minimized further without generating any coverage holes.
- (iv) Presence of a coverage hole in the neighbourhood.
- (v) Coverage hole is restored in the neighbourhood.

To achieve minimal coverage overlapping, avoid potential coverage holes and recover the holes due to node failure,

a node has to perform the following set of actions, viz., (i) Select a rescue node to recover the hole, (ii) Customize the sensing radius, (ii) Active, (ii) Sleep, (iv) Move and customize the sensing radius. Figure 5 shows possible state transitions during coverage overlap minimization and hole recovery. As WSN is a dynamic environment and the nodes are continuously learning, it is possible that a node may not take an action at all in a particular state. In Fig. 5b, the node has a set of actions available to recover the coverage hole. The action chosen by the node may vary according to the residual energy available and the energy consumed to perform an action during the exploitation phase. Afterward, as the node advances towards exploration phase, the action attributed with highest Q -value is chosen.

Algorithm 3: Game Theory-based Reinforcement Learning algorithm

Input: Q, s, a
Output: Q, r

- 1 $\lambda \leftarrow \{agt_1, agt_2, \dots, agt_n\}$, λ is a set of agents in the system.
- 2 $Sts \leftarrow \{s_1, s_2, s_3, s_4, s_5\}$, Sts is a set of states.
- 3 $A \leftarrow \{a_1, a_2, a_3, a_4, a_5, a_6\}$, A is the set of actions available to an agent.
- 4 Initialize the Q value.
- 5 **repeat**
- 6 **foreach** agt_i in a state s_i **do**
- 7 Choose an action using the ϵ -greedy based on the Q -value such that overall reward gained is maximized and Nash equilibrium is attained.
- 8 **until** the multi-agent system reaches the global optimum
- 9 Estimate the reward r gained by observing the impact on the reward parameters of the environment.
- 10 Update Q -values using Eq. (7).

4.4 Reward Estimation Criteria for MO_ZHC

The node is rewarded for its activities based on the resulting impact on the environment, both locally and globally. The targets of a node to achieve a local reward is to decrease the degree of coverage overlapping in its cell without disrupting the coverage. For this purpose, Area-wise coverage overlapping percentage $AWCOP$ estimation is done by a node for its own sensing region as well as for the AOI to determine local and global rewards, respectively. Each node finds the degree of coverage overlap in its sensing region and estimates the percentage of sensing area corresponding to each degree. Locally, each node aims to decrease the degree of coverage overlap to 0 for single degrees and higher. During this attempt, each node also makes sure that any point in its sensing area does not lose coverage. The initiator nodes in each cell collaborate with one another to estimate the $AWCOP$ with respect to four degrees (0,1,2,3) for the entire AOI. For a

region Reg in the AOI, the $AWCOP$ can be formally stated as follows for m degrees of coverage overlapping.

$$AWCOP = [Reg_0 Reg_1 Reg_2 Reg_3 \dots Reg_m] \tag{6}$$

After minimal coverage overlapping is achieved using MO_ZHC , the overlap degrees are reduced to the range $[0, 3]$ with degree 3 exceedingly small. Major percentage contributions are made by degrees 0, 1 and 2. Hence,

$$AWCOP = [Reg_0 Reg_1 Reg_2 Reg_3] \tag{7}$$

The degree of coverage overlap for any part of node’s sensing region is calculated by identifying the areas found by the intersection of the sensing ranges of the node’s first neighbours. The theorems provided in [26,27], which are completely dedicated for identifying these areas have been used in the present study. Nodes estimate the local and global rewards based on the decrease and increase observed in the value of $AWCOP$ as a result of the decisions taken by it. A node receives a local negative reward if its actions disrupt the coverage in the cell i.e. generates a coverage hole or if the existing $AWCOP$ for degrees greater than 0, increases in the cell. On the other hand, a node receives a positive local reward if the $AWCOP$ for degrees greater than 0 decreases in the AOI without generating any coverage holes. The same is the case for global rewards with respect to $AWCOP$ and coverage criteria in the AOI. A node is also provided a long-term reward based on its decision regarding the customization of the sensing range. If the energy consumed to customize the sensing range is more than the energy saved after decreasing the sensing range, then a negative reward is attributed else a positive reward is attributed to encourage the node to take such decision during similar situations in the future. The cost of increasing the sensing range by Δrs is given as follows.

$$C_s(S_j) = \begin{cases} e^{|\Delta rs_i t'|}, & \text{if } \Delta rs_i > 0 \\ -e^{-|\Delta rs_i t'|}, & \text{if } \Delta rs_i < 0 \end{cases} \tag{8}$$

where t' denotes the period starting from the point when the sensing range becomes $R_s + \Delta rs$ to the point when it is further changed by the node S_j . Negative cost is incurred when the sensing range is decreased by Δrs units. Once the overlapping is minimized, Eq. (9) is used for estimating the reward with respect to the coverage rate.

$$CR = \frac{\sum_{i=1}^{|N_a|} \pi(R_{s_i}[j])^2}{Ar} \tag{9}$$

CR represents the rate of coverage provided by the active set of nodes [13]. $|N_a|$ represents the total active nodes in the network. Ar is the area of AOI and $R_{s_i}[j]$ represents the sensing radius of i th active node during time t' . $CR > 1$

implies coverage overlapping state which means a negative reward, $CR = 1$ or CR approaching a value of 1 implies the network coverage rate is achieved which means a positive reward. If $CR < \text{threshold value}$, then it implies the presence of coverage hole in the network and yields a negative reward.

4.5 Reward Estimation Criteria for PNP_HRS

The chosen $rescue_node$ should not increase the existing coverage overlapping in the network. Choosing a $rescue_node$ also depends on the availability of sleeping nodes which decreases as the number of rounds achieved by the network increases. Negative reward is provided if there is an increase in the existing percentage of various degrees (> 0) of coverage overlapping. In PNP_HRS, along with the cost of sensing range customization, the cost of moving a node is also involved. It is estimated using Eq. (10). Δm represents the amount of distance moved by a sensor node.

$$C_P = e^{\Delta m} \tag{10}$$

Total cost of sensing range customization and node movement is calculated as follows.

$$C_T = K_1 C_P + K_2 C_s \tag{11}$$

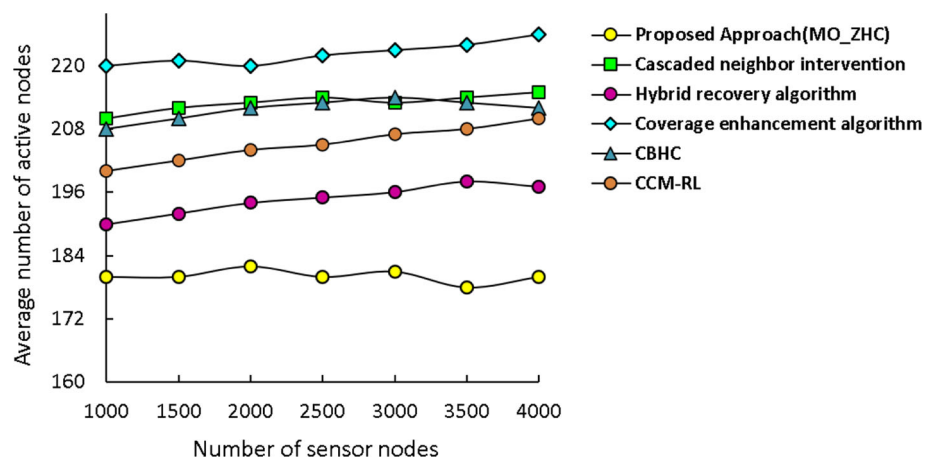
where C_s is calculated using Eq. (8), K_1 and K_2 are cost balancing coefficients for C_P and C_s , respectively. Positive rewards are provided for restoring the hole in minimum possible time which is done by comparing the $detector_node$ ’s hole restoration time with already existing minimal time of restoration. In the beginning, hole restoration time is equal to the expected time of hole restoration (t_{exp}). In this study, t_{exp} is taken as 25s for 1000 deployed nodes. (t_{exp}) is computed during the execution of PNP_HRS in a controlled environment with best and worst cases. $t_{exp} = \frac{t_b + t_w}{2}$, where t_b denotes the best case time and t_w denotes the worst case time. The best case is considered when the values of Δm and Δrs for a $rescue_node$ are equal to 0 for the hole recovery. In this case, the $rescue_node$ just needs to switch from sleep to active state after receiving a wake up signal from $detector_node$. The worst case is considered when the $rescue_node$ has to move the maximum possible distance and the value of Δrs is equal to the difference between the highest and lowest values of the sensing radius.

4.6 Time Complexity Analysis

The computational complexity of the proposed method is determined to analyse the time taken by a computer to

Table 2 Parameters and their values used in the simulation

Parameters	Values
Size of AOI	500 m × 500 m
Range of total deployed nodes	1000–4000
Initial sensing radius	25 m
Sensing range	10–35 m
Transmission range	up to 70 m
Total energy	140J
Maximum time period of a scheduling round	23 min
Learning rate	0.2
Discount factor	0.65
Average speed of a node	1 m/s

Fig. 6 Effect of total number of deployed nodes on the average number of active nodes with a comparative analysis

run the algorithm. The complexity of the proposed learning algorithm GT_RL is determined by the complexity of the algorithms proposed for overlap minimization and hole recovery, namely MO_ZHC and PNP_HRS, in the current analysis. This overall complexity is estimated to be of the order $O(Nm)$, where N is the total number of nodes deployed in the AOI and m is the average number of first neighbours a node has inside a cell partitioning the AOI.

5 Results and Discussion

This section presents the simulation results as well as the performance analysis of the proposed algorithms viz., MO_ZHC, PNP_HRS, and GT_RL. A comparative analysis is conducted between the findings of the current study and five other similar methods namely, Cascaded neighbour intervention algorithm [7], Hybrid recovery algorithm [10], Coverage enhancement algorithm [11], CBHC [12], CCM-RL [13], DHDR[15], and SOM[16]. The simulation parameters and their values are enlisted in Table 2.

5.1 Change in the Average Number of Active Nodes with Respect to Total Nodes

As the number of deployed nodes grows, Fig. 6 shows how the average number of active nodes in the network changes. The average size of the set of active nodes generated using MO_ZHC has been compared with [7,10–13]. Too many active nodes in the network incur energy wastage which can be avoided by activating only a minimal number of nodes.

The least number of average active nodes are reported for hybrid recovery algorithm [10] compared to cascaded neighbor intervention [7], coverage enhancement algorithm [11], CBHC [12], and CCM-RL [13]. However, the proposed MO_ZHC generates a significantly lower number of active nodes as compared to [10]. The average number of active nodes only rise by an average of 1.11% as the total number of nodes are increased. This is because the requirement of active nodes to cover the target area does not change if the size of the target area remains the same. The same behaviour can be observed for [7], [10–13]. The WSN is assumed to have completed one round of service when the nodes successfully transfer the sensed data to the sink node. In the current work,

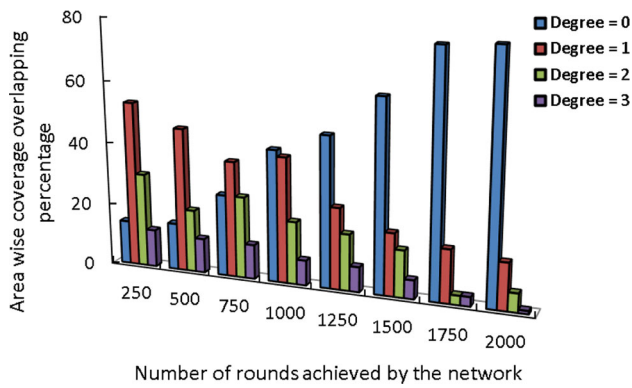


Fig. 7 Effect of the number of rounds achieved by the network on the percentage of coverage overlapping in the target area

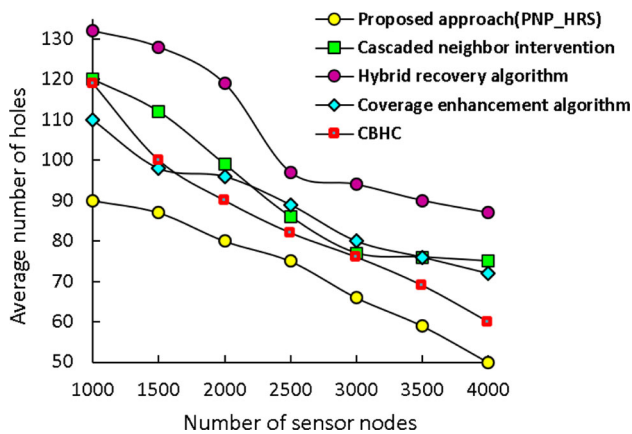


Fig. 8 Effect of the total number of deployed nodes on the average number of holes generated in the network

constrained shortest-path energy-aware routing algorithm is used for routing the data efficiently [28].

5.2 AWCOP Variation with Respect to the Number of Rounds

Figure 7 depicts the relationship between the number of rounds of service reached by the network and the AWCOP for four distinct degrees of coverage overlapping in the target region. It is essential to minimize the degrees of coverage overlapping greater than 0 to maximize the desired degree of coverage in the target area, i.e. degree 0.

A minimal coverage overlapping in the target area ensures minimal sensing data redundancy in the network. As the number of rounds reached by the network approaches 2000, the 0th degree of the coverage shows an overall increase of 400%; degree 1 shows an overall decrease of 71.6%; degree 2 shows an overall decrease of 80%, and degree 3 shows an overall reduction of 91.6%.

5.3 Variation in the Average Number of Holes with Respect to Total Nodes

The total number of coverage holes in the network changes as the number of deployed nodes varies, as shown in Fig. 8. A node has more opportunities to sleep and save energy as the number of deployed nodes in the network increases. Therefore, coverage holes due to untimely energy drainage of a node decrease. Any algorithm can only impact the holes caused by energy drainage. There is no way to monitor the holes caused by sensor node software and hardware failure before their occurrence. The proposed approach generates the least number of holes as compared to algorithms presented in [7,10–12].

When the total number of deployed nodes is increased by 500, the average decrease in the number of holes in the network is approximately 10.5%, and the overall decline in the number of holes when the total number of deployed nodes is increased from 1000 to 4000 is about 44.4%.

5.4 Changes in ANM, ANC and ANCA with Respect to Number of Rounds and Total Nodes

In Fig. 9a–d, ANM refers to the average number of nodes moved, ANC represents the average number of nodes required to customize the sensing radius and ANCA stands for the average number of nodes required to take a combined action of moving and customizing the sensing radius to restore the coverage holes. The results pertain to the proposed algorithm PNP_HRS. In Fig. 8a, for 1000 deployed nodes, the number of nodes belonging to the set ANM decrease by an average of 10.62% as the number of rounds increases by 50. This happens because an increase in the number of rounds implies that the learning of nodes has also improved towards making an optimal decision.

As the cost in terms of energy is very high for moving a node, the likeliness of such a decision decrease significantly at first. It then starts to fall nominally due to the unavailability of nodes near the coverage hole. The number of nodes belonging to the set ANC increases for the first 150 rounds and then becomes stable. The same is the case for the nodes belonging to the set ANCA. As the total deployed nodes are increased to 2000, the overall decrease in the number of nodes belonging to ANM could be determined to be 35% as the rounds rise from 50 to 300. In Fig. 9b, the total number of deployed nodes has been doubled, reducing the need to move nodes. Compared to Fig. 9a, the curve denoting the change in the number of nodes belonging to the set ANC moves up the curve depicting the number of nodes belonging to ANCA. At around 250 rounds, the number of nodes belonging to ANM and ANC is nearly equivalent. This may be due to an unexpected rise in the number of holes, necessitating the relocation of nodes once more to recover the holes. The total

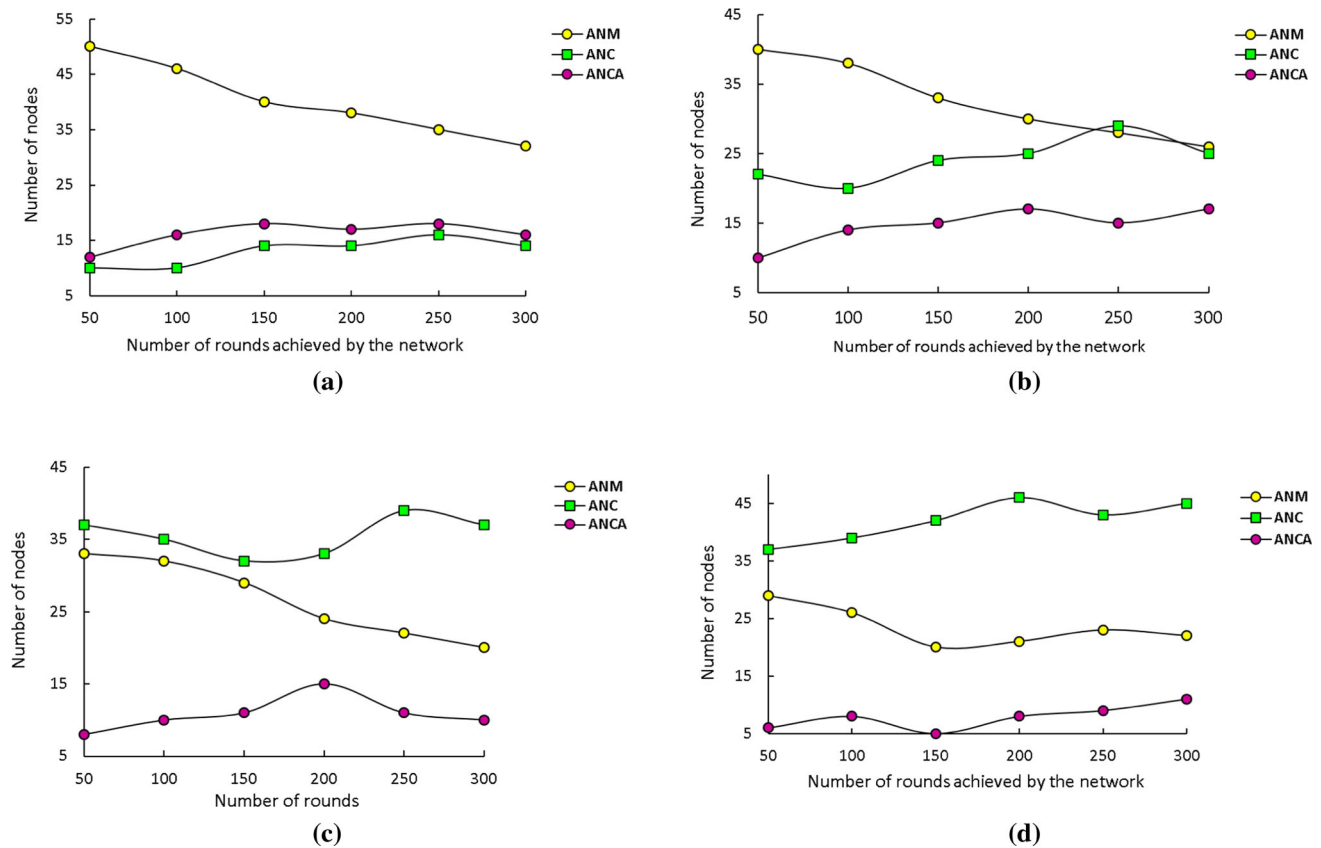


Fig. 9 Effect on the number of nodes belonging to sets ANM, ANC and ANCA with the number of rounds achieved by the network for **a** 1000; **b** 2000; **c** 3000; **d** 4000 nodes for PNP_HRS algorithm

deployed nodes in Fig. 9c are 3000, making deployment in the AOI denser. The likelihood of finding nodes near the coverage hole location has increased even more than the scenario in Fig. 9a, b. The curve representing the change in the number of nodes belonging to the set ANC has moved slightly up the curve representing the number of nodes belonging to the set ANM. The network at this point is heterogeneous, with each node having a different sensing radius based on the network requirements. The curve for nodes belonging to ANCA has moved down. This implies that the combined action of radius customization and node movement is resulting in higher costs of energy as compared to the cost pertaining to ANM. Therefore, the number of decisions related to taking a combined action has decreased. The situation similar to Fig. 9c can be observed in Fig. 9d as well. The average difference between the number of nodes belonging to set ANC and set ANM is approximately 19 nodes which show that the nodes are learning to make optimal decisions for the network.

5.5 Effect on the Average Power Consumption with Respect to Total Nodes

Figure 10 depicts the transition in the average power consumption of the active nodes as the number of deployed

nodes increases in the network. Average power consumption denotes the overall power consumed by the network during the execution of MO_ZHC, PNP_HRS and GT_RL. A comparison between the proposed approach and other existing similar works shows that the proposed method incurs the least amount of average power consumption. For the proposed solution, as the number of nodes in the network grows by 500, the power consumption increases by an average of 20.3%. One of the main reasons for this trend is that as the number of deployed nodes grows, so does the number of messages sent between them for information sharing and updating.

5.6 Average Convergence Time and the Percentage of Wrong Decisions with Respect to the Total Nodes

Figure 11 depicts the average convergence time of the learning algorithm used in the proposed solution [12,13]. Ref. [13] has the longest convergence time, while the suggested solution has the shortest. Convergence time is defined as the period in which the learning algorithm reaches its global optimum. It happens when any decision taken by a learning node is always globally optimal. Since the computations needed

Fig. 10 Effect of the number of deployed nodes on the average power consumption for the active set of nodes

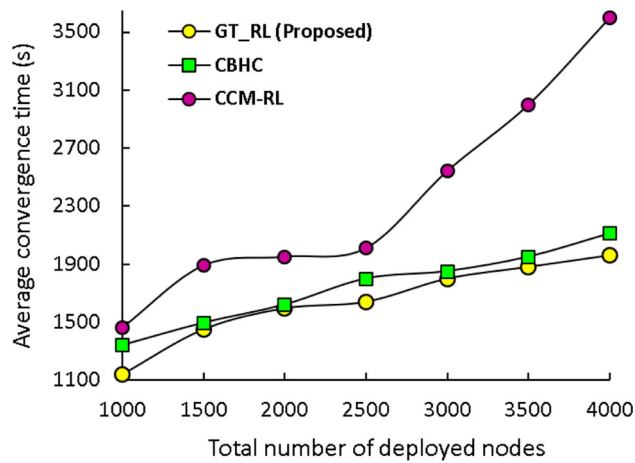
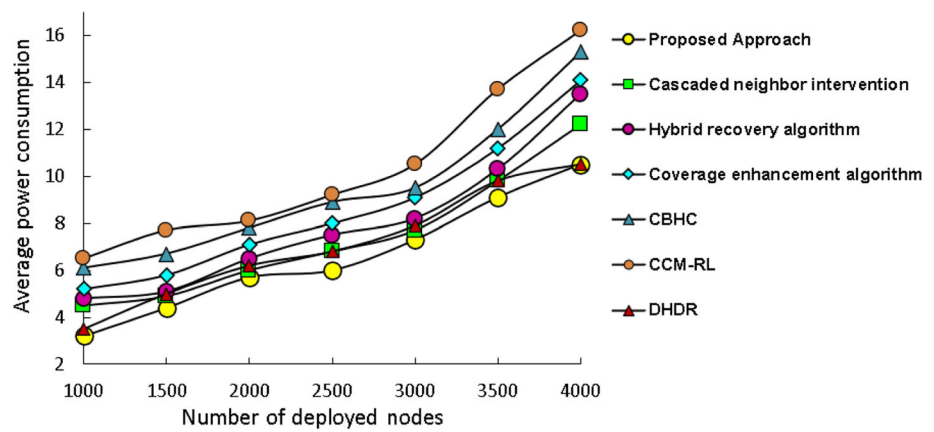


Fig. 11 Effect of total number of deployed nodes on the average convergence time for the proposed approach

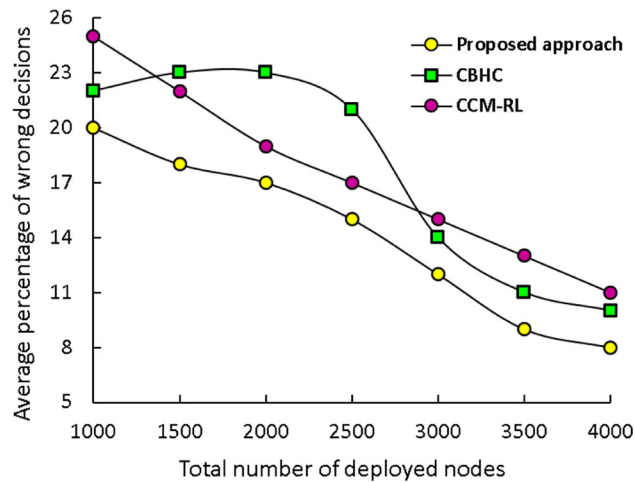


Fig. 12 Effect of total number deployed nodes on the average number of wrong decisions made by a learning node

to achieve it are energy-intensive, a minimum convergence time is desirable.

Convergence time increases with an increase in the total number of deployed nodes in the network. This is because

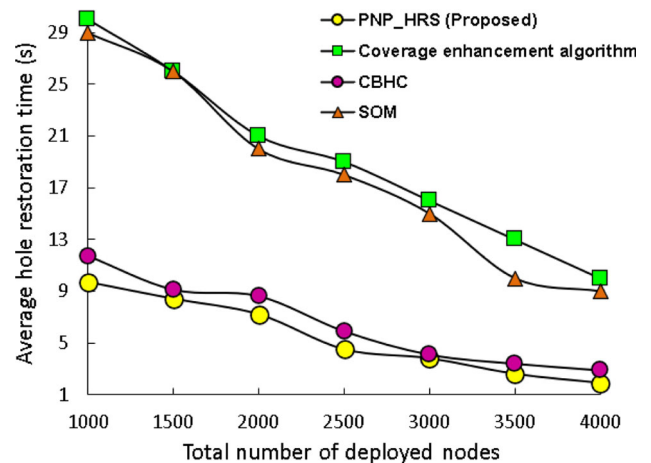


Fig. 13 Effect of total deployed nodes on the hole restoration time for proposed approach

the number of nodes influencing a node’s decision to achieve global optimality grows. When nodes are increased from 1000 to 4000, the average increase in convergence time is 9.72%. The effect of the total number of deployed nodes in the network on the average percentage of incorrect decisions can be seen in Fig. 12. While, as shown in Fig. 11, the convergence time increases as the number of nodes increases. This is because, while the convergence time for a larger number of nodes is more prolonged, a learning node has more knowledge locally and is more aware of the global optimality requirements of the network. Therefore, as the number of nodes increases, number of incorrect decisions made by an agent before reaching the convergence point decreases.

5.7 Variation in the Average Hole Restoration Time and Distance Moved by a Node with Respect to Total Nodes

For the proposed hole recovery algorithm PNP_HRS, coverage enhancement algorithm [11] and CBHC [12], Fig. 13 depicts the impact of the total number of nodes on the average

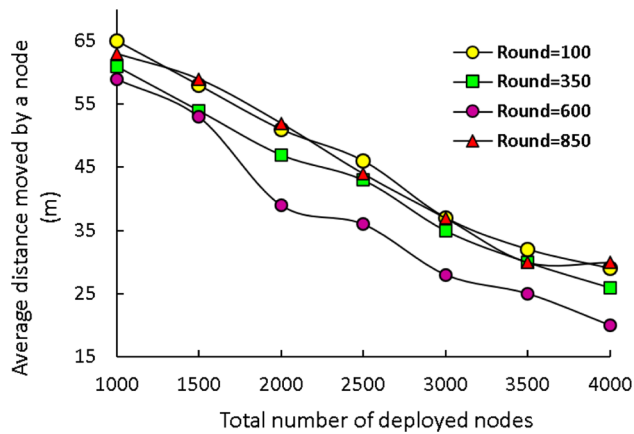


Fig. 14 Effect of total deployed nodes on the average distance moved by a node

hole restoration period. The cumulative time from when the detector node finds the hole and when the rescue node restores the coverage hole is referred to as hole restoration time. PNP_HRS shows the least restoration time. When nodes are increased in the range from 1000 to 4000, the average reduction in restoration time is estimated to be 20.5%. For different values of rounds reached by the network, Fig. 14 shows the effect of the total number of deployed nodes on the average distance moved by the node. As the number of rounds increases from 100 to 600, the curve depicting the change in average length traversed by a node to repair a hole, decreases with the total number of deployed nodes. But when the network reaches the 850th round, the curve moves up. This is because as the network reaches higher rounds of service, the death rate of the nodes increases, reducing the availability of nodes near the coverage holes.

6 Conclusion

The current study focuses on network coverage holes and the overlapped coverage issue in a WSN. The target area is divided into square cells so that nodes in different cells can learn simultaneously without interfering with one another's decisions. A Minimal Overlapping with Zero Holes Coverage algorithm is introduced to minimize the coverage overlapping due to the random deployment of sensor nodes. The nodes aim to minimize the Area Wise Percentage of Coverage Overlapping Percentage for degrees ≥ 1 without generating new holes. While reducing the coverage overlap, boundary nodes also make sure that no coverage holes are generated near the cell boundaries due to simultaneous learning. A second algorithm called Predictable, and Non-Predictable Holes Recovery Scheme is designed to recover the network from predictable and non-predictable coverage holes. The active nodes select a sleeping node to recover

a detected coverage hole, using the least amount of energy in the shortest amount of time possible. The utilization of sleeping nodes allows the network to maintain the minimal coverage overlapping and guarantees that no new holes shall be generated during the hole recovery process. Game Theory-based Reinforcement Learning algorithm is proposed to help the nodes learn from their actions. GT_RL incorporates MO_ZHC and PNP_HRS so that nodes can take actions strategically during their initial stage of learning rather than taking random actions. Each node strives to achieve Nash Equilibrium and the state of global optimum. Once GT_RL converges, each node becomes capable of making the best decisions independently. The simulation results are recorded, and a comparison with the cascaded neighbor intervention algorithm [7], the hybrid recovery algorithm [10], the coverage enhancement algorithm [11], the CBHC [12], and the CCM-RL [13] is performed. Various performance metrics, namely the average number of active nodes, area wise coverage overlapping percentage, the average number of holes, the average number of nodes moved, the average number of nodes moved with sensing range customization, the average number of nodes with sensing range customization for the hole recovery, average power consumption, average convergence time, percentage of wrong decisions, average hole restoration time, and average distance moved by a node to recover a hole, have been evaluated, which verify the efficacy of the proposed algorithms. It shall be interesting to investigate the occurrence of coverage holes during random deployment. Exploration of the integration of the WSN with the 5G networks. The effect of 5G networks on the communication overhead and the security aspects of WSN can also be considered for future study.

Author Contributions Nilanshi Chauhan developed the concept, conducted the analysis, and generated the results. Piyush Rawat also helped in conducting the analysis. Dr. Siddhartha Chauhan oversaw the research work and validated the findings.

Funding The authors gratefully acknowledge financial support from the MoE, Govt. of India, New Delhi, for the first author.

Availability of Data and Material All the data generated and analysed during this study is included in the article. No other source was used.

Declarations

Conflict of interest The authors declare that they have no established conflicting financial interests or personal relationships that may have influenced the research presented in this paper.

References

1. Akyildiz, I.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E.: Wireless sensor networks: a survey. *Comput. Netw.* **38**(4), 393–422 (2002)

2. Zhouzhou, L.; She, Y.: Hybrid wireless sensor network coverage holes restoring algorithm. *J. Sens.* **2016**, 1–10 (2016)
3. Sahoo, P.K.; Liao, W.C.: HORA: a distributed coverage hole repair algorithm for wireless sensor networks. *IEEE Trans. Mob. Comput.* **14**(7), 1397–1410 (2015)
4. Abolhasan, M.; Maali, Y.; Rafiei, A.; Ni, W.: Distributed hybrid coverage hole recovery in wireless sensor networks. *IEEE Sens. J.* **16**(23), 8640–8648 (2016)
5. Aliouane, L.; Benchaiba, M.: HACH: healing algorithm of coverage hole in a wireless sensor network. In: *Proceedings of 8th International Conference on Next Generation Mobile Applications*, pp. 215–220 (2014)
6. Amgoth, T.; Jana, P.K.: Coverage hole detection and restoration algorithm for wireless sensor networks. *Peer-to-Peer Netw. Appl.* **10**, 66–78 (2017)
7. Khalifa, B.; Aghbari, Z.A.; Khedr, A.M.; Abawajy, J.H.: Coverage hole repair in WSNs using cascaded neighbor intervention. *IEEE Sens. J.* **17**(21), 7209–7216 (2017)
8. Verma, M.; Sharma, S.: A greedy approach for coverage hole detection and restoration in wireless sensor networks. *Wirel. Pers. Commun.* **101**(1), 75–86 (2018)
9. Khedr, A.M.; Osamy, W.; Salim, A.: Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks. *Comput. Commun.* **124**, 61–75 (2018)
10. Hajje, F.; Hamdi, M.; Ejbali, R.; Zaied, M.: A distributed coverage hole recovery approach based on reinforcement learning for wireless sensor networks. *Ad Hoc Netw.* **101**, 1–16 (2020)
11. Priyadarshi, R.; Gupta, B.: Coverage area enhancement in wireless sensor network. *Microsyst. Technol.* **26**, 1417–1426 (2020)
12. Singh, P.; Chen, Y.C.: Sensing coverage hole identification and coverage hole healing methods for wireless sensor networks. *Wirel. Netw.* **26**, 2223–2239 (2019)
13. Sharma, A.; Chauhan, S.: A distributed reinforcement learning based sensor node scheduling algorithm for coverage and connectivity maintenance in wireless sensor network. *Wirel. Netw.* **26**, 4411–4429 (2020)
14. Khalifa, B.; Al Aghbari, Z.; Khedr, A.M.: A distributed self-healing coverage hole detection and repair scheme for mobile wireless sensor networks. *Sustain. Comput. Inform. Syst.* **66**, 1–10 (2020)
15. Khalifa, B.; Al Aghbari, Z.; Khedr, M.A.: A distributed self-healing coverage hole detection and repair scheme for mobile wireless sensor networks. *Sustain. Comput. Inform. Syst.* **30**, 100428 (2021). <https://doi.org/10.1016/j.suscom.2020.100428>
16. Yang, C.; Su, S.; Ju, X.; Song, J.: A mobile sensors dispatch scheme based on improved SOM algorithm for coverage hole healing. *IEEE Sens. J.* **21**(18), 1–10 (2021). <https://doi.org/10.1109/JSEN.2021.3095511>
17. El Khamlichi, Y.; Mesmoudi, Y.; Tahiri, A.; Abtoy, A.: A recovery algorithm to detect and repair coverage holes in wireless sensor network systems. *J. Commun.* **13**(2), 67–74 (2018). <https://doi.org/10.12720/jcm.13.2.67-74>
18. Singh, P.; Chen, Y.C.: Sensing coverage hole identification and coverage hole healing methods for wireless sensor networks. *Wirel. Netw.* **26**(12), 2223–2239 (2019). <https://doi.org/10.1007/s11276-019-02067-7>
19. Liu, Y.; Yang, Z.: *Location, Localization, and Localizability: Location-Awareness Technology for Wireless Networks*. Springer (2010)
20. Chauhan, N.; Chauhan, S.: A novel area coverage technique for maximizing the wireless sensor network lifetime. *Arab. J. Sci. Eng.* **46**, 3329–3343 (2021)
21. Soni, S.; Shrivastava, M.: Novel learning algorithms for efficient mobile sink data collection using reinforcement learning in wireless sensor network. *Wirel. Commun. Mob. Comput.* **2018**, 1–13 (2018)
22. Mustapha, S.; Abdelhamid, M.; Amar, A.: Random deployment of wireless sensor networks: a survey and approach. *Int. J. Ad Hoc Ubiqu. Comput.* **15**, 133–146 (2014). <https://doi.org/10.1504/IJAHUC.2014.059905>
23. Kozłowski, M.; McConville, R.; Santos-Rodríguez, R.; Piechocki, R.: *Energy Efficiency in Reinforcement Learning for Wireless Sensor Networks*, pp. 1–16 (2018)
24. Nowé, A.; Vrancx, P.; De Hauwere, Y.M.: *Game Theory and Multi-agent Reinforcement Learning in Reinforcement Learning*, pp. 441–470. Springer, Berlin (2012)
25. Watkins, C.J.C.H.; Dayan, P.: Technical note: Q-learning. *Mach. Learn.* **8**, 279–292 (1992)
26. Hawbani, A.; Wang, X.; Kuhlani, H.; Ghannami, A.; Farooq, M.U.; Al-sharabi, Y.: Extracting the overlapped subregions in wireless sensor networks. *Wirel. Netw.* **25**, 4705–4726 (2019)
27. Librino, F.; Levorato, M.; Zorzi, M.: An algorithmic solution for computing circle intersection areas and its applications to wireless communications. In: *Proceedings of the 7th International Conference on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pp. 239–248 (2009)
28. Youssef, M.; Younis, M.; Arisha, K.: A constrained shortestpath energy-aware routing algorithm for wireless sensor networks. In: *Proceedings of the IEEE Wireless Communication and Networks Conference*, pp. 1–12 (2002)

