



A localization and deployment model for wireless sensor networks using arithmetic optimization algorithm

Soumya J Bhat¹ · Santhosh K V¹

Received: 15 September 2021 / Accepted: 8 February 2022 / Published online: 7 March 2022
© The Author(s) 2022

Abstract

With the advancement of sensor technologies, Wireless Sensor Networks (WSN) are envisioning a rich variety of promising services in many fields. WSN is formed by the deployment of sensor nodes in the regions of interest using a deterministic or random deployment strategy. The random deployment strategy is more suitable in large monitoring areas and harsh environments. But, in this type of deployment, coverage holes and disconnected networks can exist. Important events may get unnoticed reducing the reliability of the networks. Since the locations of nodes are unknown in a randomly deployed WSN, it is difficult to locate the holes. For this, we are proposing a localization and deployment model. The localization algorithm uses Arithmetic Optimization Algorithm (AOA) and the results of this algorithm are further used to develop a deployment model to achieve a completely connected network. This algorithm is tested in various fields. The algorithm is able to localize nodes accurately and identify the coverage holes with an error rate of less than 0.27% when the Average Localization Error (ALE) is within 5m.

Keywords Arithmetic optimization algorithm · Deployment · Heterogeneous · Range free localization · Wireless sensor networks

1 Introduction

Wireless Sensor Networks (WSN) have gained popularity in recent years because of their utilization in numerous monitoring and control applications. These are simple networks formed by a group of sensor nodes that communicate with each other through wireless channels [1–3].

WSNs are used in various applications such as oil and gas explorations, intrusion detection, military surveillance, forest fire monitoring, precision agriculture, and road traffic tracking [4–6]. The deployment strategy of sensor nodes varies depending on the application. This can be broadly classified as deterministic deployment and random deployment [7]. The deterministic deployment is more suitable for

friendly environments with a limited sensor area. But, in applications like oil and gas explorations, military surveillance, natural disasters, forest fire monitoring, sensor nodes need to be deployed in large areas [8]. Sometimes, the region of deployment is harsh and unreachable. In such cases, manual placement of nodes in fixed locations is impractical [9]. Hence, random deployment of sensor nodes is carried out in the required fields of interest using an aerial vehicle. But, in this type of deployment, there is no guarantee of complete coverage and connectivity. Sensor nodes may cluster in some regions leaving out coverage holes and disconnected networks in other regions [10]. These coverage holes and disconnected networks result in missing out on important events and hence reduces the quality of the network [11]. Coverage holes and disconnected networks can also exist because of the presence of obstacles, energy dissipation of nodes in certain regions, radio signal attenuation and jamming [12]. The operator needs to know about such conditions to take further action. Many solutions are developed for this purpose. To address the problem of coverage holes in WSN with static sensor nodes, solutions based on the properties of the Delaunay triangle, Hamilton cycle, perimeter coverage of sensing range, etc are reported [13–15].

✉ Santhosh K V
kv.santhu@gmail.com

Soumya J Bhat
soumya.janardhana@gmail.com

¹ Department of Instrumentation and Control Engineering,
Centre for Cyber Physical System, Manipal Institute
of Technology, Manipal Academy of Higher Education,
Manipal 576104, India

The existing deployment algorithms assume that sensor nodes are aware of their locations [16]. But, in reality, as sensor nodes are randomly deployed, they are unaware of their locations. Attaching every sensor node with positioning devices such as Global Positioning System (GPS) receivers is not possible because of increased cost, higher energy consumption and lower accuracy due to shadowing [17]. Hence, to estimate the locations of sensor nodes in random deployment, localization algorithms are used [18]. But, the locations obtained from localization algorithms are only estimated locations with errors [19]. These errors impact the efficiency of the deployment algorithms. To improve localization accuracy, we have reported an Arithmetic Optimization Algorithm based Localization (AOAL) for the localization of sensor nodes with minimal errors. The localization problem is formed as an optimization problem and solved using Arithmetic Optimization Algorithm (AOA) [20]. To the best of our knowledge, AOA has been utilized for the first time to solve the problem of localization. Also, the distance measurement among nodes is enhanced by utilizing the communication radius of nodes. We have observed improved localization results compared to other algorithms. Further, we have proposed a deployment model called Coverage Hole Discovery (CHD) for static nodes using the localization results of the AOAL algorithm. In this deployment method, sensor nodes use the information collected for the localization process such as the neighbor list and their estimated locations to identify the coverage holes in a distributed manner. The main contributions of this paper are highlighted as follows:

- A novel AOAL algorithm for the localization of sensor nodes using connectivity information.
- A deployment model utilizing the information collected for localization purpose to achieve a completely connected network.
- Analysis of the impact of localization errors on the developed deployment algorithm.
- Comparison of the reported method along with different localization algorithms.

The rest of the paper is organized as follows. In Sect. 2, the related localization and deployment methods are reviewed. The network model is discussed in Sect. 3 and the developed localization and deployment model is elaborated in Sect. 4. Simulation results are discussed in Sect. 5. Finally, Sect. 6 provides the conclusion and proposed future work.

2 Related works

The deployment strategy determines the reliability of the WSN in terms of its ability to monitor, collect and transmit the required data. Researchers have reported different

deployment methods to enhance the performance of the WSN in various applications.

Various deterministic deployment plans are reported in [21–25] for the applications of precision agriculture, smart grid, indoor environments, structural health monitoring systems, etc. Deterministic deployment methods offer optimum network configuration, but they are more suitable for small scale WSN. Also, this type of deployment is impractical and impossible in large scale WSN and in inaccessible and harsh regions of interest. Generally, in such conditions, random deployment is the only available option. [26] reported a solution to the WSN deployment problem by hybridizing two meta-heuristics, namely the Bees algorithm and the Grasshopper optimization algorithm. Here, the strength of Grasshopper optimization is used to enhance the exploitation phase of the Bees algorithm. [27] reported a sensor deployment technique based on the space between the sensor node and its neighboring sensors to improve the coverage area after a random deployment. This method capitalizes on the coverage of the detected area by forcing the sensor to change its position toward the area with a lower sensor density. But, these approaches are not relevant for static sensors which are unable to change their initial positions.

However, static sensors are widely used in real life network applications to reduce costs and save energy [28]. To identify coverage holes after random deployment in a static WSN, [15] reported a method in which each sensor node determines if every segment of its sensing perimeter is covered by its neighbors. This method assumes the locations and sensing neighbors are known to the nodes. The algorithm is analyzed in 2D square fields. [13] reported a coverage hole detection method based on the properties of empty circles. Empty circles are the circumcircle of the Delaunay triangle. Coverage holes are identified by comparing the radius of the empty circle with the radius of the sensing range. [14] reported a distributed algorithm with only connectivity information to detect coverage holes in a static WSN. Here, a neighbor graph is constructed based on the neighbor information. Then, each node determines whether it can be a hole boundary or not by checking for the existence of the Hamilton cycle in its neighbor graph. The results are analyzed in 2D square fields.

Most of the coverage hole discovery algorithms for static WSN use location and neighbor node information. But, in random deployment, the locations of nodes are unknown. Localization algorithms are developed for this purpose. These algorithms use few location known reference nodes and localize other nodes with reference to these nodes. In recent years, various localization algorithms are developed considering the practical challenges of the deployment environment such as 3D fields, anisotropy, effects of obstacles on the distance measurements, etc. [29] reported a localization algorithm using augmented Lagrangian matrix to fill

and reconstruct the complete Euclidean distance matrix and establish a complete mathematical model. Another localization algorithm called SL-Free localization is reported in [30]. This method consists of three steps namely relative localization, geographical distances evaluation and the fine localization for the accurate positioning of nodes. [31] reported MDS-MAP(D) positioning algorithm based on steepest descent optimization method. A range free localization algorithm based on Harris Hawks optimization method is reported in [32]. A multihop range-free localization method for anisotropic networks is reported in [33]. Here the distance estimation between sensor and reference nodes is improved using a distributed pattern-driven scheme. [17] reported an adaptive flower pollination algorithm with enhanced exploration and exploitation capabilities for the localization of sensor nodes in WSN. [34] reported another localization algorithm using the mean-shift clustering method for outlier detection in a shadowed environment.

In this paper, we use a simple and straightforward optimization algorithm called AOA for the localization of sensor nodes along with a novel distance estimation technique utilizing the communication range of nodes to improve the localization accuracy of the nodes. The results are further used to develop a deployment method to achieve a connected network.

3 Network model

A WSN of N sensor nodes is considered to be deployed randomly in the required field of interest. Let R be a small fraction of sensor nodes that are attached with GPS receivers. These nodes are aware of their locations and act as reference nodes in localization algorithms. After deployment, nodes communicate with other nodes located in their communication region and form an adhoc network.

The communication range of nodes is defined using log normal path loss model [18]. Let (x_i, y_i) be the location, CR_{N_i} be the communication range and S_{N_i} be the sensing range of node N_i . The communication range of a node can be obtained as,

$$CR_N = 10^{\left(\frac{P_t - P_{th}}{10\eta}\right)} \quad (1)$$

where P_t is the transmitted power, P_{th} is the minimum threshold receiver power required by nodes, and η is the path loss factor. The nodes in the network start monitoring the events occurring in their sensing area and report them to the network operator through multihop communication.

Sensing model The sensing area of a sensor node N_i is the area in which N_i can detect the occurrence of events. In practice, because of environmental factors, the sensing area of a

node is of irregular shape. We have defined the sensing range S_{N_i} , as the lower bound of the irregular sensing area of nodes [13]. Here, all the events occurring within the distance S_{N_i} from node N_i are determined with probability 1 [15]. We have assumed $CR_{N_i} \geq 2 \times S_{N_i}$.

Heterogeneity The nodes are assumed to be heterogeneous with different transmission powers and sensing ranges. This means, CR_{N_i} and CR_{N_j} need not be same for any $i, j; i \neq j$.

Problem description When the sensors are deployed randomly in hostile target fields, nodes may cluster in some regions and leave holes in other regions. Even if care is taken to deploy nodes uniformly, a coverage hole can still appear because nodes can be destroyed, or the signal may be jammed. The holes can cause network disconnection. Important events may not get detected and thereby reducing the reliability of the networks. Therefore, it is important to detect and heal the holes in the network. Figure 1 shows few examples of the coverage holes.

In a randomly deployed network, nodes collect information from neighbors and localize themselves. The nodes are localized with respect to the locations of reference nodes. When they observe any event, they report it to the operator along with location information. Accurate location of nodes is important to identify the location of occurrence of events and to take action on them. This location information is also important to identify coverage holes and disconnected networks in a randomly deployed network. Hence, we have reported a method where nodes can localize themselves in a distributed manner and then identify the holes by themselves and report them to the operator.

4 Details of the localization and deployment plan

To overcome the above mentioned challenges, in this section, a new localization and deployment model is developed to achieve a completely connected network. Here, initially, nodes are deployed randomly. Then, nodes are localized by estimating their distance from reference nodes using the sum of communication ranges in the shortest path and evaluating the locations using AOA. Further, nodes uncover the coverage holes and disconnected networks by analyzing the distribution of neighbor nodes around them.

The steps are described here.

4.1 Initial deployment

WSN is formed by the random deployment of sensor and reference nodes. The node density should be selected carefully

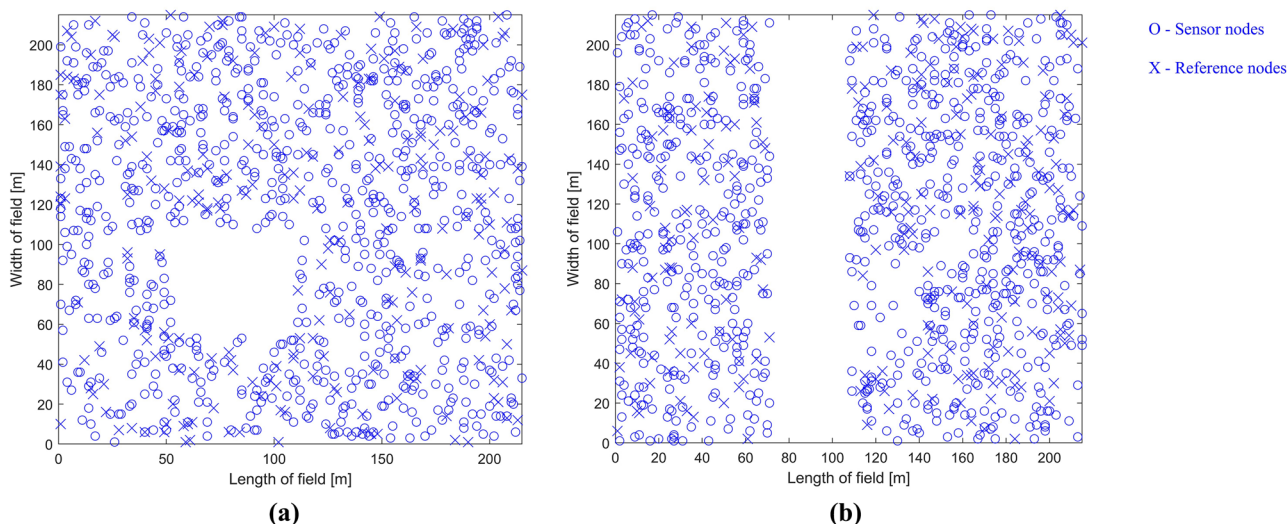


Fig. 1 (a) Coverage hole (b) Disconnected network in a 2D square field

to achieve the required level of coverage. As discussed in [35], the required node density to achieve k -coverage is given in Eq. (1).

$$node\ density = \frac{k}{0.422 \times r_0^3} \tag{2}$$

where, $r_0 = S_{min}/1.066$. S_{min} is the minimum sensing radius among the heterogeneous set of nodes. All the nodes are initialized as non-Border with a parameter *Border* set as 0.

4.2 Localization of nodes

After the initial deployment, nodes localize themselves using the AOAL method. The initial step is similar to hop based algorithm, but the hop counts are replaced by communication ranges. This change helps in overcoming the impact of heterogeneity on distance estimations. Here, every reference node broadcasts the location information (x, y) and communication range CR to the neighbor nodes. The neighbor nodes store this information, add their CR value to the received CR . If this sum of CR is less than a predefined $CR_{threshold}$, nodes broadcast them along with location information of reference nodes. The $CR_{threshold}$ helps to identify only the closer reference nodes, thus reducing the impact of irregular bent paths on distance estimations. After receiving this information, neighbor nodes verify if the obtained reference node information is already stored in their neighbor information. If it is a new node information, the neighbor information is updated with the received reference node location and sum of CR . If the obtained node information is already stored, the received sum of CR is compared with the stored value and

the minimum among these two CR values is saved with location information. At the end of this step, every node will have location information of reference nodes and the corresponding minimum value of the sum of CR from reference node to itself.

For example, consider a WSN shown in Fig. 2. Here, neighbor information stored in $N1$ is: $NSET_1 = \{1. (x_{R1}, y_{R1}), CR_{R1}, 2. (x_{R2}, y_{R2}), CR_{R2}, 3. (x_{R3}, y_{R3}), \min (CR_{R3} + CR_{R1}, CR_{R3} + CR_{N2})\}$.

After this, reference nodes in the network measure the Average Communication Distance (ACD) as a ratio of the distance between reference nodes and the sum of CR between them. ACD for a reference node R_1 is defined as

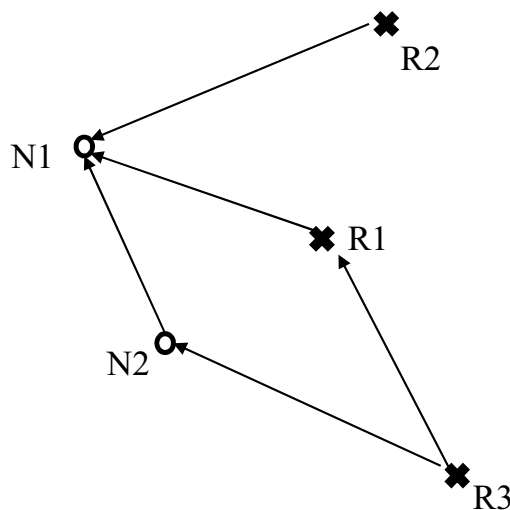


Fig. 2 An example of a WSN

$$ACD_{R1} = \frac{\sum_{i \in NSET_1} d_{R1Ri}}{\sum_{i \in NSET_1} CR_{R1Ri}} \quad (3)$$

where d_{R1Ri} is the distance between two nodes $R1$ and Ri .

Reference nodes broadcast the measured ACD to other nodes. Sensor nodes estimate the distance to reference nodes as the product of received ACD and the sum of communication ranges CR . The distance between sensor node Ni and reference node Rj is measured as,

$$\tilde{d}_{NiRj} = ACD_{Rj} \times CR_{NiRj} \quad (4)$$

The obtained distances are only estimated values affected by errors due to irregularities in the field. Next, the location estimation problem is formulated as an optimization problem and solved using these distance measurements.

The optimization problem: $\forall i, 1 \leq i \leq N$, find location estimation $(\tilde{x}_i, \tilde{y}_i)$ as a point in the field such that the distance from $(\tilde{x}_i, \tilde{y}_i)$ to every reference node $j \in NSET_i$ is equal to \tilde{d}_{ij} .

Or equivalently,

$$\forall i; 1 \leq i \leq N, \text{ find } (\tilde{x}_i, \tilde{y}_i) \text{ s.t. } |\sqrt{(\tilde{x}_i - x_j)^2 + (\tilde{y}_i - y_j)^2} - \tilde{d}_{ij}|$$

is a minimum for every $j \in NSET_i$.

The defined localization problem is solved using AOA. AOA is a population-based algorithm consisting of exploration and exploitation phases. This algorithm is developed by considering simple arithmetic operators (i.e., Multiplication, Division, Subtraction, and Addition) to determine the best element subjected to specific criteria from a set of candidate alternatives [20].

This algorithm takes population size P , maximum number of iterations T , upper bound UB , lower bound LB as input values and searches for a best solution by measuring cost value.

This is implemented as follows. To localize an unknown node i , input parameters $LB = (LB_x, LB_y)$ and $UB = (UB_x, UB_y)$ are set as arrays of lower and upper boundary values of the deployment field. AOA generates initial P locations randomly within the boundaries LB and UB . The cost of each of these P location estimations is measured as below:

$$COST = \sum_{j \in NSET_i} |\sqrt{(\tilde{x}_i - x_j)^2 + (\tilde{y}_i - y_j)^2} - \tilde{d}_{ij}| \quad (5)$$

The location with minimum cost value is saved as $P_{BEST} = (\tilde{x}_i, \tilde{y}_i)$.

Previously generated P location estimates are next modified in exploration and exploitation phases of AOA to converge towards the actual location of node i . This is done using Math Optimizer Accelerator (MOA) and Math Optimizer Probability (MOP) which are defined as a function of current iteration t as below:

$$MOA(t) = 0.2 + t \times \frac{0.8}{T} \quad (6)$$

$$MOP(t) = 1 - \frac{t^{0.2}}{T^{0.2}} \quad (7)$$

Generate random numbers r_1, r_2 and r_3 between 0 and 1.

If $r_1 < MOA$, modify $\tilde{x}_{i,k}, 1 \leq k \leq P$ as

$$\tilde{x}_{i,k}(t) = \begin{cases} \frac{\tilde{x}_{i,k}(t-1)}{(MOP(t)+\epsilon)} \times ((UB_x - LB_x) \times 0.5 + LB_x) & \text{if } r_2 > 0.5 \\ \tilde{x}_{i,k}(t-1) \times MOP(t) \times ((UB_x - LB_x) \times 0.5 + LB_x) & \text{otherwise} \end{cases} \quad (8)$$

Here, ϵ is a small number set as $2.2204e - 16$.

Else, if $r_1 \geq MOA$, modify $\tilde{x}_{i,k}, 1 \leq k \leq P$ as

$$\tilde{x}_{i,k}(t) = \begin{cases} \tilde{x}_{i,k}(t-1) - MOP(t) \times ((UB_x - LB_x) \times 0.5 + LB_x) & \text{if } r_3 > 0.5 \\ \tilde{x}_{i,k}(t-1) + MOP(t) \times ((UB_x - LB_x) \times 0.5 + LB_x) & \text{otherwise} \end{cases} \quad (9)$$

Similarly, value of $\tilde{y}_{i,k}(t)$ is modified. This process is continued to get the best location with minimum cost in T iterations. The final solution $P_{BEST} = (\tilde{x}_i, \tilde{y}_i)$ is considered as the location estimate of the unknown node.

4.3 Coverage hole discovery (CHD)

After localization, every node is aware of its location in the field. Next, this location information is used to discover coverage holes. This is done by checking for uniform distribution of neighbor nodes around every node with overlapping sensing ranges. For this, every node identifies few points evenly spaced around it at a distance of S_{Ni} . The number of points identified can be any number greater than 1. After identifying the points, the nodes check if these points are inside the sensing range of any of the neighbor nodes stored in the neighbor matrix. If any one of the identified points of a node doesn't reside inside the sensing range of any of its neighbor nodes, the node is identified as a border node with the *Border* parameter set to 1. By selecting more points, even coverage holes with smaller sides can be detected. But, since every point needs to be checked for overlapping neighbor nodes, this also increases the computation overhead. Hence, depending on the need of the application, suitable number of points need to be identified.

This step is explained with an example. Here, we have considered 8 evenly spaced points. In this case, the coverage holes with sides greater than $(1/8)^{th}$ of the perimeter of the sensing area can be detected. Consider a node N_1 having N_2, \dots, N_9 as neighbors stored in its neighbor matrix. To check if N_1 is in the boundary of any coverage hole, it first identifies 8 points, $P1_{N1}, P2_{N1}, \dots, P8_{N1}$ as shown in Fig. 3. Next, every point is checked for an overlapping sensing range.

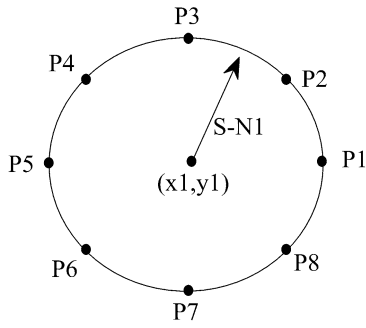


Fig. 3 Points on the sensing range of node N1

For node N_1 , if there is any N_i where $i = 2, 3, \dots, 9$ such that $d_{(P_{jN_1})-N_i} \leq (S_{N_i}) \forall j = 1, \dots, 8$, then the node N_1 is marked as not a border node with $Border=0$. Here, $d_{(P_{jN_1})-N_i}$ indicates the distance between (P_{jN_1}) and N_i . If any P_{jN_1} does not find a N_i such that $d_{(P_{jN_1})-N_i} \leq (S_{N_i})$, then the node N_1 is marked as a border node with $Border=1$. This can be a node at the border of a field or at the border of a coverage hole. As shown in Fig. 4, point P_5 of node N_1 is not covered by sensing region of any

of its neighbor nodes. Hence, node N_1 is identified as a border node.

4.4 Update the end user

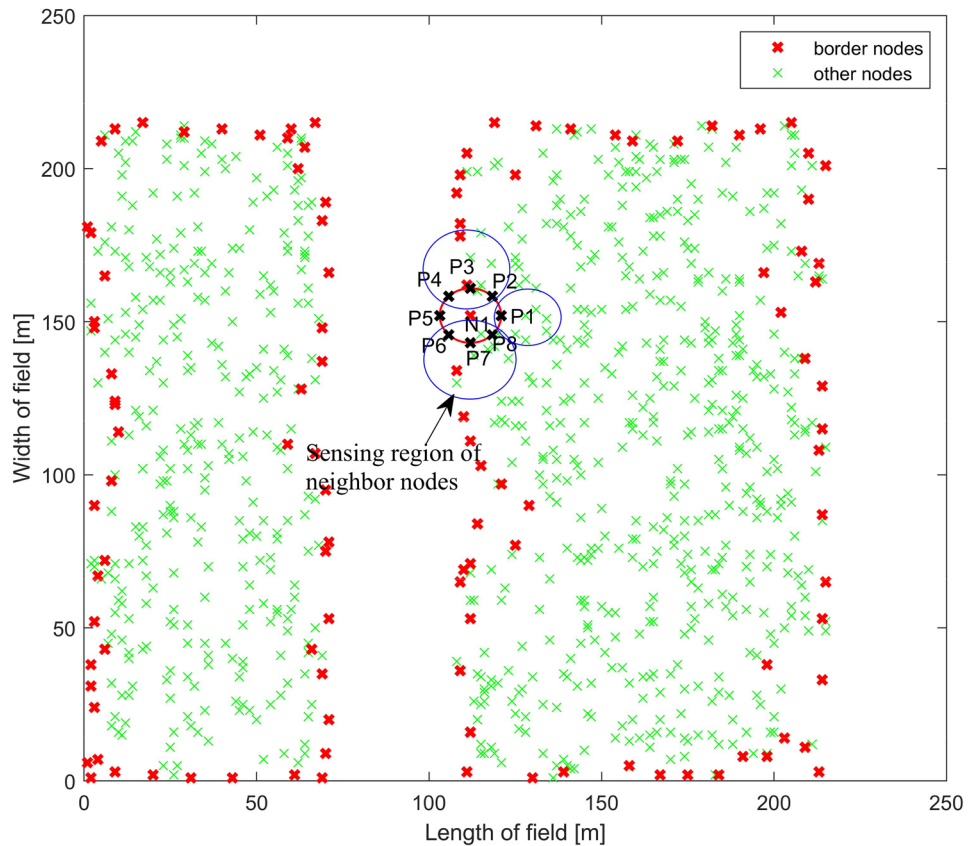
When the node detects any change in its border status, it alerts the end user. This way, the user will know all the border nodes at the borders of coverage holes and fields. He can take further actions based on the requirements.

4.5 Disconnected networks

As proved in [36], if the coverage radius of a node is at least twice the sensing range, a completely covered network will be a completely connected network. Hence, by ensuring complete coverage, we get a completely connected network.

The pseudo-code of the reported localization and deployment method is shown in Algorithm 1.

Fig. 4 An example of a border node identification



Algorithm 1: Localization and deployment method

Input: Sensor nodes, reference nodes, field dimensions and the required node density.

Output: A WSN with complete coverage and connectivity.

1. Find the required number of nodes to achieve the desired node density using Eq.2.
 2. Carry out uniform random distribution of nodes in the required field of interest.
 3. Initialize the parameter $Border=0$.
 4. Reference nodes broadcast their location information and CR to other nodes in the network.
 5. Every sensor node stores the minimum sum of CR within $CR_{threshold}$ along with location information of reference nodes.
 6. Reference nodes measure ACD using Eq.3 and broadcast this to other nodes.
 7. Sensor nodes measure the distance to reference nodes using Eq.4.
 8. Sensor nodes estimate their locations using AOA from Eq.5 through Eq.9.
 9. Localized sensor nodes broadcast the location information to other nodes in the network.
 10. Sensor nodes update the neighbor matrix.
 11. Every node identifies 8 points on the perimeter of its sensing range.
 12. for Point = 1 to 8
 - if distance from Point to every neighbor node > sensing range of the neighbor node
 - $Border = 1$.
 - Exit for.
 - else
 - $Border = 0$.
 - end if
 13. end for
 14. All the nodes whose Border status is changed will alert the end user through other nodes.
 15. User can initiate random or deterministic deployment in the identified region.
-

4.6 Computational complexity

The computational complexity of the proposed model relies on two factors: localization and deployment. Localization is achieved by AOA. The computational complexity of the AOA is of $O(P \times (2 \times T + 1))$ where P is the population size, T is the iterations. The computational complexity of the deployment model is $O(8 \times AN)$ where AN is the average number of immediate neighbors for a node in the network.

In the next section, the proposed model is evaluated under various scenarios.

5 Results and analysis

In this section, the reported localization and deployment model is evaluated by comparing the results with the traditional Centroid [37] and the recent range-free algorithms

EWCL [38] and HHO-AM [32]. For this, a combination of isotropic and anisotropic fields are considered. These fields are simulated by defining the required boundary shapes using the MATLAB R2018a tool. Sensor and reference nodes are randomly deployed in these fields. This is done by generating locations for nodes within the boundary region using the built in uniform random distribution function of MATLAB. Sensor nodes are considered to be heterogeneous with sensing ranges varying from 6m to 12m and communication ranges varying from 12m to 24m. After deployment, nodes communicate with each other and form a network. The transmission of signals among nodes undergoes attenuation in space, which is represented by the path loss factor of 4. Apart from path loss, the presence of obstacles causes attenuation of signals in different directions, resulting in irregular communication patterns. The degree of irregularity in the communication field is assumed to be 0.005. The irregular communication patterns are generated using weibull random distribution as suggested in [39].

After deployment, localization algorithms are used to estimate the locations of sensor nodes by considering the actual locations of reference nodes. The localization algorithms are then evaluated by comparing the actual locations of sensor nodes with their estimated locations. This is done using the parameter Average Localization Error (ALE) [40] which is defined as,

$$ALE = \frac{\sum_{i=1}^n \sqrt{(\tilde{x}_i - x_i)^2 + (\tilde{y}_i - y_i)^2 + (\tilde{z}_i - z_i)^2}}{n} \quad (10)$$

where (x_i, y_i, z_i) is the actual node location, $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ is the measured node location and n is the number of localized nodes.

After localization, nodes identify themselves as border or non-border nodes using the reported method. The efficiency of this is evaluated by analyzing the percentage of non-border nodes erroneously reporting as border nodes. The deployment error is measured as,

Deployment error (%)

$$= \frac{\text{Count of erroneously reported border nodes}}{\text{Nodes in the network}} \times 100 \quad (11)$$

5.1 Evaluation of AOAL

The reported AOAL is evaluated under various scenarios.

5.1.1 Influence of node density

Here, the performance of the reported algorithm is evaluated under various node densities. For this, sensor and reference nodes are deployed in a square shaped field. The count of

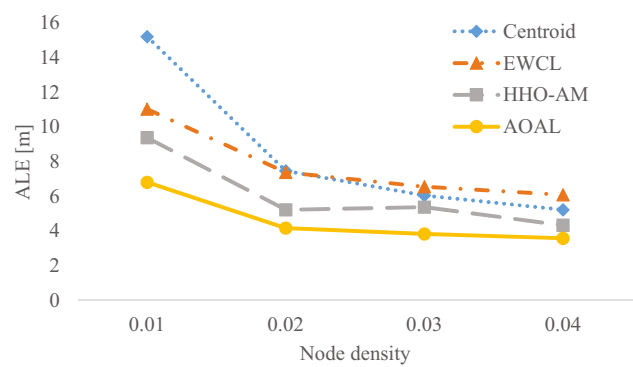


Fig. 5 Comparison of localization error at varying node densities

nodes is increased from 500 to 2000. Reference node density is maintained at 20%. The ALE at node densities varying from $0.01/m^2$ to $0.04/m^2$ is compared with other localization algorithms. Comparison of localization results for different algorithms is shown in Fig. 5.

From Fig. 5, it can be observed that the localization error of the Centroid algorithm is very high at lower node densities whereas AOAL has shown the least localization error. With an increase in node density, the performance of localization algorithms improves. AOAL has shown an overall improvement of 24% in ALE compared to HHO-AM.

5.1.2 Influence of reference node density

The localization algorithm is evaluated at different reference node ratios. Node density is fixed at $0.03/m^2$ and reference node density is varied from 10% to 40%. The localization errors from different algorithms are shown in Fig. 6.

While the error rate was very high for other localization algorithms at lower reference node densities, AOAL has performed well. It has shown an overall improvement of 24 – 41% when compared with Centroid, EWCL, and HHO-AM localization algorithms.

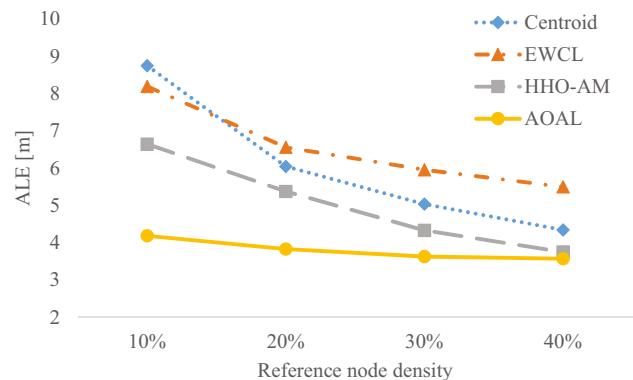


Fig. 6 Comparison of localization error at varying reference node densities

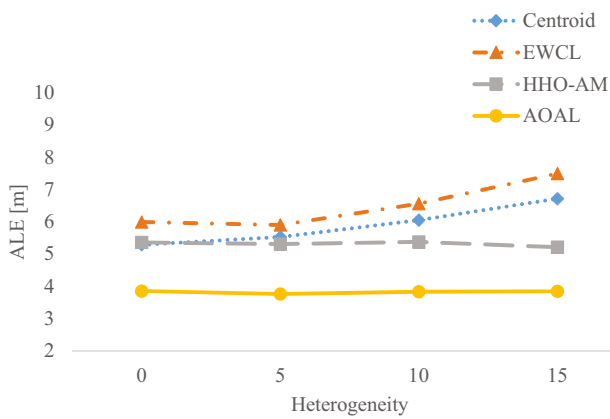


Fig. 7 Comparison of localization error at varying transmission powers

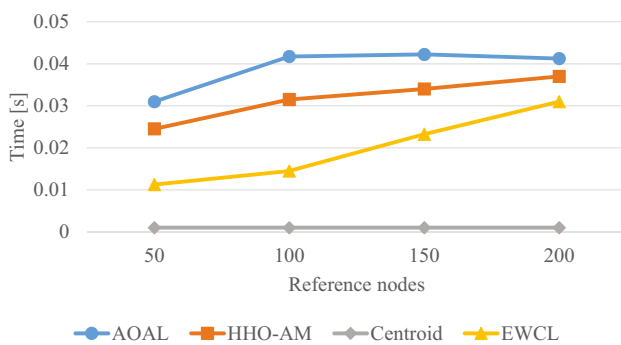


Fig. 8 Comparison of computational time

5.1.3 Influence of heterogeneity

Here, the behavior of the localization algorithms at various heterogeneous conditions is analyzed. For this, sensor nodes of different transmission powers are considered. The deviation in transmission powers of nodes is varied from 0 to 15dBm

and the localization errors are observed in these situations. Figure 7 shows the localization errors in these conditions.

The increase in heterogeneity deteriorated the performance of Centroid and EWCL. But, HHO-AM and AOAL showed more stable performance in heterogeneous conditions. The localization accuracy of AOAL is improved by 28% compared to the HHO-AM algorithm.

5.1.4 Computational time

In this section, the complexity of the localization algorithm is evaluated in terms of computational time. The average computational time for localization of a node is compared with other localization algorithms and the results are provided in Fig. 8.

The computational time of Centroid algorithm is the least as it considers only immediate neighbor nodes for location analysis. The EWCL and HHO-AM localization algorithms consider reference nodes within a threshold distance, their computational time is higher than Centroid algorithm. But, computational time of AOAL algorithm is observed to be higher than HHO-AM and EWCL.

5.2 Evaluation of the deployment model

After localization, nodes identify the coverage holes and disconnected networks using the reported deployment model.

5.3 Field anisotropy

To evaluate the effectiveness of the reported deployment method on various types of fields, a combination of sensor and reference nodes are randomly deployed in a square field with a disconnected network and a C shaped field with a hole as shown in Figs. 9a and 10a. Node density is maintained at

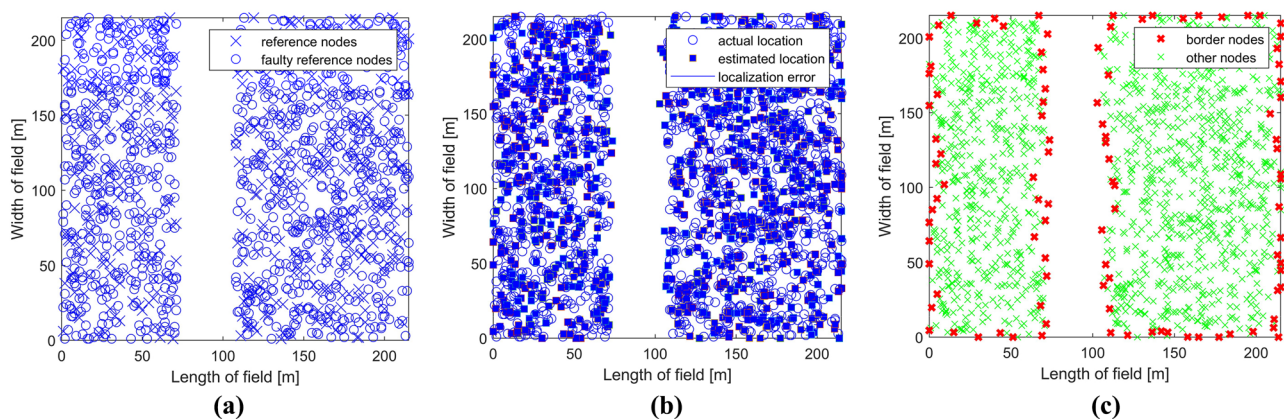


Fig. 9 Square field: (a) Random deployment, (b) Localization results and (c) Coverage hole discovery

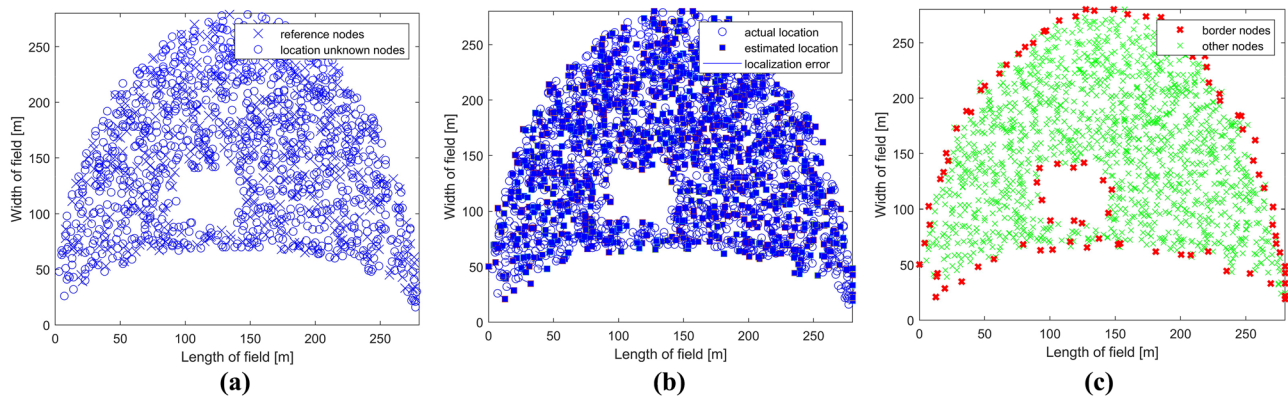


Fig. 10 C shaped field: (a) Random deployment, (b) Localization results and (c) Coverage hole discovery

Table 1 Comparison of deployment error with ALE

Field type	Node density	Centroid		EWCL		HHO-AM		AOAL	
		ALE in(m)	Error in (%)	ALE in(m)	Error in (%)	ALE in(m)	Error in (%)	ALE in(m)	Error in (%)
2D square field with disconnected network	0.01	10.47	10.19	9.86	8.24	9.72	4.12	6.05	1.52
	0.02	5.98	1.83	8.28	1.51	5.69	0.43	4.16	0.22
	0.03	5.0207	0.57	6.79	0.00	4.80	0.07	3.73	0.07
2D C field with a hole	0.04	4.49	0.05	6.81	0.00	4.10	0.00	3.55	0.00
	0.01	14.13	8.02	10.36	3.90	8.08	0.86	6.96	1.73
	0.02	6.08	2.48	6.82	0.75	6.32	0.21	5.03	0.21
	0.03	5.10	1.01	5.76	0.50	4.28	0.14	4.76	0.00
	0.04	4.94	0.27	6.01	0.00	3.91	0.00	4.52	0.00

$0.03/m^2$. After the initial random deployment, sensor nodes localize themselves using a localization algorithm. Here, we have used the AOAL localization algorithm. The results of the localization algorithm are shown in Figs. 9b and 10b. After localization, sensor nodes use their estimated locations and their neighbor nodes to identify holes. The results of the coverage hole discovery method are shown in Figs. 9c and 10c. The nodes identified as border nodes are marked in red color and non-border nodes are marked in green color. We can observe that using the proposed method, nodes were able to identify the coverage holes and disconnected networks effectively.

5.4 Influence of node density and localization errors

To study the influence of node density on the reported deployment algorithm, initial random deployment is carried out by varying the node density from $0.01/m^2$ to $0.04/m^2$. Then, the nodes are localized using different localization algorithms namely Centroid, EWCL, HHO-AM, and AOAL. The results of ALE and deployment error at various node densities in two different fields are shown in table 1. Figure 11 shows the measured deployment

error in a square field when the nodes are localized using Centroid, EWCL, HHO-AM, and AOAL localization algorithms. Border node detection errors of sensor nodes are in the range of 2 – 10% at $0.01/m^2$ node density. This is because the number of neighbor nodes available for node localization is small which causes higher localization errors. Higher localization errors resulted in higher errors in border node identification. With an increase in node density, the ALE of localization algorithms is reduced and hence node discovery errors are also reduced.

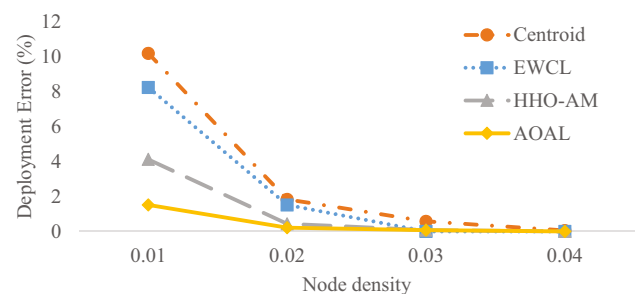
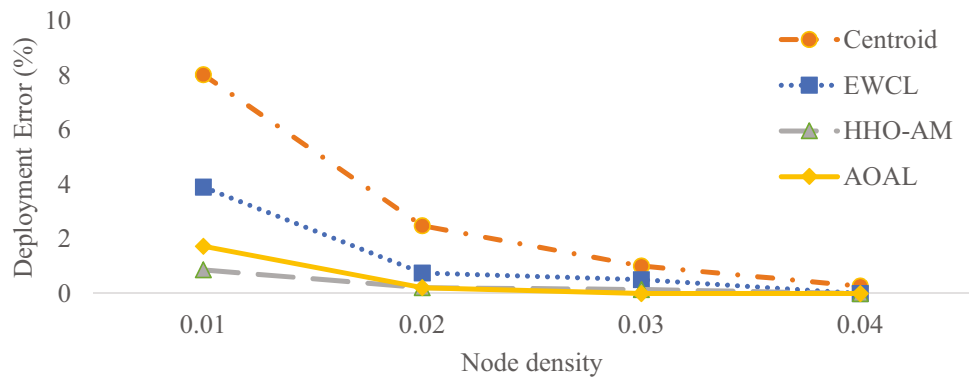


Fig. 11 Influence of node density on deployment error in a square field

Fig. 12 Influence of node density on deployment error in a C field



Similar results are observed when the tests are repeated in a C shaped field with a coverage hole. The results are shown in Fig. 12. When the node density is increased from 0.01% to 0.04%, the average ALE from four localization algorithms is reduced from 10m to 5m. This reduced the average deployment errors from 3.6% to 0.07%.

6 Conclusions

WSNs have recently attracted considerable research interest by providing unprecedented opportunities for monitoring and controlling applications. WSN can be formed by following deterministic or random deployment strategies. When sensor nodes are deployed using a random deployment strategy, sensor nodes may cluster in some regions leaving out coverage holes and disconnected networks in other regions. But, since in random deployment, the user is not aware of the locations of the deployed nodes, it is not possible to identify the holes. To overcome this problem, we have reported a localization and deployment plan to help the end user in identifying the coverage holes and disconnected networks in a distributed manner. The reported method uses AOA to identify the locations of the nodes. The localized nodes then use their neighbor information collected for localization purposes to identify missing coverage regions. This method is tested in various fields for different node densities. AOAL showed an improvement of 24% to 45% in ALE compared to other methods. It is also observed that by reducing the average localization errors from 10m to 5m, the average deployment errors were also reduced from 5% to 0.04%. In the future, the algorithm will be enhanced to reduce deployment errors even in lower node densities.

Funding information Open access funding provided by Manipal Academy of Higher Education, Manipal. No funding was received for conducting this work.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Zameni M, Rezaei A, Farzinavash L (2021) Two-phase node deployment for target coverage in rechargeable WSNs using genetic algorithm and integer linear programming. *J Supercomput* 77(4):4172–4200. <https://doi.org/10.1007/s11227-020-03431-7>
- Deniz F, Bagci H, Korpeoglu I (2021) Energy-efficient and fault-tolerant drone-BS placement in heterogeneous wireless sensor networks. *Wireless Netw* 27(1):825–838. <https://doi.org/10.1007/s11276-020-02494-x>
- Ullah A, Said G, Sher M, Ning H (2020) Fog-assisted secure healthcare data aggregation scheme in IoT-enabled WSN. *Peer-to-Peer Networking and Applications* 13(1):163–174. <https://doi.org/10.1007/s12083-019-00745-z>
- Savazzi S, Spagnolini U, Goratti L, Molteni D, Latva-aho M, Nicoli M (2013) Ultra-wide band sensor networks in oil and gas explorations. *IEEE Commun Mag* 51(4):150–160. <https://doi.org/10.1109/MCOM.2013.6495774>
- Zhu C, Zheng C, Shu L, Han G (2012) A survey on coverage and connectivity issues in wireless sensor networks. *J Netw Comput Appl* 35(2):619–632. <https://doi.org/10.1016/j.jnca.2011.11.016>
- Singh RK, Aernouts M, De Meyer M, Weyn M, Berkvens R (2020) Leveraging LoRaWAN technology for precision agriculture in greenhouses. *Sensors* 20(7):1827. <https://doi.org/10.3390/s20071827>

7. Amutha J, Sharma S, Nagar J (2020) WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues. *Wireless Pers Commun* 111(2):1089–1115. <https://doi.org/10.1007/s11277-019-06903-z>
8. Annepu V, Rajesh A (2019) Implementation of self adaptive mutation factor and cross-over probability based differential evolution algorithm for node localization in wireless sensor networks. *Evol Intel* 12(3):469–478. <https://doi.org/10.1007/s12065-019-00239-0>
9. Otero CE, Shaw WH, Kostanic I, Otero LD (2010) Multiresponse optimization of stochastic WSN deployment using response surface methodology and desirability functions. *IEEE Syst J* 4(1):39–48. <https://doi.org/10.1109/JSYST.2009.2039737>
10. Du Y (2020) Method for the Optimal Sensor Deployment of WSNs in 3D Terrain Based on the DPSOVF Algorithm. *IEEE Access* 8:140806–140821. <https://doi.org/10.1109/ACCESS.2020.3013106>
11. Xing Y, Huangfu W, Dai X, Hu X (2017) Node Localization Based on Multiple Radio Transmission Power Levels for Wireless Sensor Networks. In *International Conference on 5G for Future Wireless Networks* (pp. 19–27). Springer, Cham. https://doi.org/10.1007/978-3-319-72823-0_3
12. Ma HC, Sahoo PK, Chen YW (2011) Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks. *J Netw Comput Appl* 34(5):1743–1756. <https://doi.org/10.1016/j.jnca.2011.06.007>
13. Li W (2014) A novel graphic coverage hole description in wireless sensor networks. *IEEE Commun Lett* 18(12):2205–2208. <https://doi.org/10.1109/LCOMM.2014.2366094>
14. 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* (pp. 1–6). IEEE. <https://doi.org/10.1109/GLOCOM.2011.6134128>
15. Khedr AM, Osamy W, Salim A (2018) Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks. *Comput Commun* 124:61–75. <https://doi.org/10.1016/j.comcom.2018.04.002>
16. Yarinezhad R, Hashemi SN (2020) A sensor deployment approach for target coverage problem in wireless sensor networks. *J Ambient Int Humanized Comp*. 1–16. <https://doi.org/10.1007/s12652-020-02195-5>
17. Singh P, Mittal N (2021) An efficient localization approach to locate sensor nodes in 3D wireless sensor networks using adaptive flower pollination algorithm. *Wireless Netw* 27(3):1999–2014. <https://doi.org/10.1007/s11276-021-02557-7>
18. Bhat SJ, Santhosh KV (2020) Is Localization of Wireless Sensor Networks in Irregular Fields a Challenge? *Wireless Pers Commun* 114:2017–2042. <https://doi.org/10.1007/s11277-020-07460-6>
19. Bhat SJ, Santhosh KV (2021) A Method for Fault Tolerant Localization of Heterogeneous Wireless Sensor Networks. *IEEE Access* 9:37054–37063. <https://doi.org/10.1109/ACCESS.2021.3063160>
20. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng* 376:113609. <https://doi.org/10.1016/j.cma.2020.113609>
21. Qiao Y, Dao TK, Pan JS, Chu SC (2020) Diversity teams in soccer league competition algorithm for wireless sensor network deployment problem. *Symmetry* 12(3):445. <https://doi.org/10.3390/sym12030445>
22. Priyadarshi R, Gupta B, Anurag A (2020) Wireless sensor networks deployment: a result oriented analysis. *Wireless Pers Commun* 113(2):843–866. <https://doi.org/10.1007/s11277-020-07255-9>
23. Garcia L, Parra L, Jimenez JM, Parra M, Lloret J, Mauri PV, Lorenz P (2021) Deployment Strategies of Soil Monitoring WSN for Precision Agriculture Irrigation Scheduling in Rural Areas. *Sensors* 21(5):1693. <https://doi.org/10.3390/s21051693>
24. Sarobin MVR (2020) Optimized Node Deployment in Wireless Sensor Network for Smart Grid Application. *Wireless Pers Commun* 111(3):1431–1451. <https://doi.org/10.1007/s11277-019-06925-7>
25. Bhuiyan MZA, Wang G, Cao J, Wu J (2013) Deploying wireless sensor networks with fault-tolerance for structural health monitoring. *IEEE Trans Comput* 64(2):382–395. <https://doi.org/10.1109/TC.2013.195>
26. Deghbouch H, Debbat F (2021) A Hybrid Bees Algorithm with Grasshopper Optimization Algorithm for Optimal Deployment of Wireless Sensor Networks. *Inteligencia Artificial*, 24(67):18–35. <https://doi.org/10.4114/intartif.vol24iss67pp18-35>
27. Sateesh VA, Kumar A, Priyadarshi R, Nath V (2021) A novel deployment scheme to enhance the coverage in wireless sensor network. In *Proceedings of the Fourth International Conference on Microelectronics, Computing and Communication Systems* (pp. 985–993). Springer, Singapore. <https://doi.org/10.1007/978-981-15-5546-682>
28. Wang G, Guo L, Duan H, Liu L, Wang H (2012) Dynamic deployment of wireless sensor networks by biogeography based optimization algorithm. *J Sens Actuator Netw* 1(2):86–96. <https://doi.org/10.3390/jsan1020086>
29. Wang P, Tu G (2020) Localization algorithm of wireless sensor network based on matrix reconstruction. *Comput Commun* 154:216–222. <https://doi.org/10.1016/j.comcom.2020.01.051>
30. Silmi S, Doukha Z, Moussaoui S (2021) A self-localization range free protocol for wireless sensor networks. *Peer-to-Peer Networking and Applications*, 1–11. <https://doi.org/10.1007/s12083-021-01155-w>
31. Teng Z, Qian L, Huang J (2021) Multi-target localization algorithm for wireless sensor network. *Peer-to-Peer Networking and Applications*, 1–8. <https://doi.org/10.1007/s12083-021-01193-4>
32. Bhat SJ, Venkata SK (2020) An optimization based localization with area minimization for heterogeneous wireless sensor networks in anisotropic fields. *Comput Netw* 179:107371. <https://doi.org/10.1016/j.comnet.2020.107371>
33. Xiao Q, Xiao B, Cao J, Wang J (2010) Multihop range-free localization in anisotropic wireless sensor networks: A pattern-driven scheme. *IEEE Trans Mob Comput* 9(11):1592–1607. <https://doi.org/10.1109/TMC.2010.129>
34. Chuku N, Nasipuri A (2021) RSSI-Based localization schemes for wireless sensor networks using outlier detection. *J Sens Actuator Netw* 10(1):10. <https://doi.org/10.3390/jsan10010010>
35. Ammari HM, Das S (2009) A study of k-coverage and measures of connectivity in 3D wireless sensor networks. *IEEE Trans Comput* 59(2):243–257. <https://doi.org/10.1109/TC.2009.166>
36. Zhang H, Hou JC (2005) Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc Sens. Wirel. Networks* 1(1–2):89–124
37. Bulusu N, Heidemann J, Estrin D (2000) GPS-less low-cost outdoor localization for very small devices. *IEEE Pers Commun* 7(5):28–34. <https://doi.org/10.1109/98.878533>
38. Kaur A, Kumar P, Gupta GP (2019) A weighted centroid localization algorithm for randomly deployed wireless sensor networks.

J King Saud University-Computer Info Sci 31(1):82–91. <https://doi.org/10.1016/j.jksuci.2017.01.007>

39. Zhou G, He T, Krishnamurthy S, Stankovic JA (2006) Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 2(2):221–262. <https://doi.org/10.1145/1149283.1149287>
40. Raguraman P, Ramasundaram M, Balakrishnan V (2018) Localization in wireless sensor networks: A dimension based pruning approach in 3D environments. *Appl Soft Comput* 68:219–232. <https://doi.org/10.1016/j.asoc.2018.03.039>

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



Soumya J. Bhat received Bachelor of Engineering degree from Manipal Institute of Technology, India and Master of Technology from Malnad College of Engineering, India. She has pursued internship at Honeywell Technology Solutions Lab Pvt. Ltd. for 1 year and worked as R&D engineer in Nokia Solutions and Networks India Private Limited for 9 years. Currently, she is a research scholar at the Department of Instrumentation and Control Engineering, Manipal Institute of Technology,

India. Her research interests include wireless sensor networks, wireless communication, network planning and localization.



Dr. Santhosh KV received his bachelors and Masters from Visvesvaraya Technological University, Belgaum in 2004 and 2006. He received his PhD from National Institute of Technology, Silchar in the year 2013. Dr. Santhosh KV is currently working as Associate Professor, in the Department of Instrumentation and Control Engineering, Manipal Institute of Technology, MAHE. Dr. Santhosh KV research interest are in the area of sensor fusion, smart sensor, and fault detection and isolation.