AN EFFICIENT APPROACH FOR COVERAGE HOLE DETECTION AND HEALING IN WSN.

Saroj Devi[1], Dr. Priyanka Bansal[2], Dr. Mukesh Singla[3]

[1]Research scholar, ECE Deptt., Baba Mastnath University, Rohtak,

[2]Professor, ECE Deptt., Baba Mastnath University, Rohtak,

[3]Dean, Faculty of Engineering, Baba Mastnath University, Rohtak,

(devisaroj6940@gmail.com)

A Novel Method for WSN Mitigation Coverage Hole

Abstract

One of the biggest side advantages of modern large-scale mobile networks and their inherent wireless sensors is the ability to conduct extensive, thorough monitoring. Nevertheless, sensor nodes in a WSN are often deployed at random, which may result in inefficiencies in terms of dependability and quality of service due to Coverage Holes and extensive overlap in the regions being monitored. With WSN, the coverage gaps issue just involves optimizing the available resources. Nevertheless, coverage gaps cannot be eliminated entirely in WSN, and their prevalence will increase until a remedy is discovered. As a result, fixing coverage gaps is essential for the effective and efficient operation of the sensor network as a whole. Using the Delaunay technique and an enhanced Arithmetic Optimization Algorithm, this research suggests a new approach to identifying and fixing coverage gaps in Wireless Sensor Networks (WSNs) (IAOA). With this strategy, redundant nodes are strategically placed throughout the network to cover all possible regions. The proposed IAOA includes a more effective exploration phase that quickly pinpoints where the redundant nodes should be placed. Results reveal that compared to the state-of-the-art PSOCGSA (Hybrid optimisation) technique, the suggested methodology may enhance coverage hole mitigation by as much as 59.15%. The findings prove that the suggested method is useful for improving WSN performance.

Keywords: *Wireless Sensor Networks and Mitigation of Coverage Hole.*

Introduction

As such methods as Wireless Sensor Networks collect and manage data utilising numerous small intelligent devices in dispersed ad hoc networks, which may even be formed in inaccessible and dangerous places, to monitor many applications, they have acquired increasing popularity in recent years. The lifetime of a network is increased, and congestion is reduced, thanks to self-organization and cooperation between nodes. Without wireless connectivity, computer networks would be unable to provide essential services like transportation. WSNs are used for a wide variety of purposes, including but not limited to environmental and real-world habitat monitoring, machine surveillance, precision farming, interior management, intelligent alarms, and military applications. To detect and communicate environmental or physical parameters, such networks utilise a large number of electronically dispersed sensors over the area of interest (AoI). Sensor nodes in WSN are initially placed at random, however this may lead to inefficiencies in efficiency, reliability, and quality of service due to Coverage Holes and large overlapping areas in the regions being monitored. Coverage gaps in a WSN environment may be caused by a number of factors, including but not limited to sporadic deployment, sensor battery exhaustion, hardware failures, software mistakes, and security threats.

There are several factors, such as node failure, communication range constraints, and environmental impediments that may lead to coverage gaps in wireless sensor networks (WSNs). Each coverage gap, regardless of its cause, degrades network performance in the same way by shortening network life expectancy or even causing complete network collapse. Hence, it is essential for better network performance and reliability to identify and address coverage gaps in WSNs. For instance, RSS-based methods estimate coverage gaps by analysing the signal intensity information received from neighbouring nodes. Nevertheless, coverage gaps may be estimated indirectly via the use of network topology data like node density and connection. A few examples of indirect techniques include the Delaunay triangulation, the Voronoi diagram and the k-coverage. When a gap in coverage is found, the following step is to fill it. Using geometry, we can construct a network of triangles where no one point is inside the circle of any other triangle. Simply said, the Delaunay Triangulation guarantees that every triangle is of good form and does not overlap with any others. The use of Delaunay Triangulation in wireless sensor networks (WSNs) is only one of several. In order to fill up the coverage gap, these nodes are often deployed in the immediate area. Energy limits, node placement constraints, and interference all make it difficult to pinpoint the sweet spot for redundant nodes. Many optimisation methods, such as genetic algorithms and particle swarm optimisation, have been offered as potential solutions to this problem.

Literature Survey

Metaheuristics are a kind of sophisticated mathematical optimisation approach used to search for a solution, especially when either a large amount of data or computational resources are unavailable. [14] The "genetic algorithm" and its less effective cousin, the "greedy approach," are two very effective methods. The evolutionary algorithm not only provides a globally optimum solution but also requires fewer sensor nodes to maintain connection. Nevertheless, the greedy method provides a locally optimum solution at the expense of requiring a larger number of relay nodes to enable K- connection. Particle swarm optimisation was developed by Eberhart and Kennedy and is inspired by the coordinated behaviour of flocks of birds. To show how well PSO works with the probability sensing model, we ran simulations in MATLAB. Inspired by the beehive swarm's foraging area, Artificial Bee Colony (ABC) does the same action repeatedly to solve the problem of continuous optimisation. Although this method improves performance and expands coverage in WSN, it does not take device connectivity to the base station into consideration. Through graph transformation, the probability-based Ant Colony Optimization (ACO) approach transforms the original issue into a pathfinding problem. In comparison to GPRS, this method consumes less time and resources while providing the network with a dynamic implementation that fairly considers all relevant characteristics while selecting the optimal route from among the shortest ones.

With WSN, the coverage gaps issue just involves optimising the available resources. However, coverage gaps cannot be eliminated entirely in WSN at this time, and their prevalence will increase if no remedy is developed. As a result, fixing coverage gaps is essential for the effective and efficient operation of the sensor network as a whole. Since gaps in coverage are inevitable, it's helpful to have a strategy that can rapidly and precisely identify such gaps and fill them in. Nevertheless, pinpointing the coverage gap in real-time testing is quite minimal. In various contexts, such as healthcare monitoring, supply chain management and thermal analysis, approaches based on swarm intelligence (SI) provide a superior answer. Even terminal illnesses have been proposed to be diagnosed using these nature-based theories. The amount of heat lost by the thermal receiver tube may be calculated using an algorithm called Integrated PSO (iPSO), which proved that SI-based techniques were superior to other optimisation algorithms inspired by nature.

As a result, the given literature works to define that geometrically-based hole-finding algorithms achieve excellent accuracy. While there are many different methods for plugging a hole, the most majority of them include some kind of iterative improvement. These optimisation approaches address the hole-healing issue by relocating the auxiliary nodes to the location of the coverage gap. Yet, in recent decades, the complexity and intricacy of real-world challenges have risen, calling for more reliable optimisation approaches, especially meta-heuristic optimisation algorithms. Real-world optimisation problems are already challenging because of their non-linear constraints, complexity, computing expense, lack of convexity, and large search areas. For many optimisation problems, these stochastic approaches provide approximations of optimal solutions. Yet, conventional optimisation techniques have been supplanted by gradient-free approaches that are very effective in avoiding local optima. The optimal set of variables for a function may be found by minimising or optimising its aim. The two main meta-heuristic optimisation search approaches are exploration/diversification and exploitation/intensification. International searching is an integral part of exploration, and it helps avoid and overcome the problem of being stuck at a local optimum. Exploitation, on the other hand, entails focusing on improving quality at a more regional level. Maintaining a happy medium between these two strategies is crucial to the efficacy of any algorithm. These characteristics are used by numerous operators and procedures in all population-based algorithms. Theoretical studies in the literature, however, show that algorithms' relative prominence often shifts as older ones evolve, hybridise, and give way to more recent ones. Evolutionary algorithms, swarm intelligence, and physics-based methods all have several practical applications. The No Free Lunch theorem, however, argues that no one optimisation strategy can address all optimisation challenges; hence, researchers do not rely on a singular approach. This study proposes implementing a reworked arithmetic optimisation algorithm in order to deal with recent and emerging difficulties (IAOA).

Research Contributions

The Voronoi diagram and Delaunay Triangulation are used to further improve the method's realisation. The Voronoi diagram is a set of non-overlapping polygons that represent the network regions closest to some particular sensor nodes, and the Delaunay triangulation is a set of non-overlapping triangles that connect sensor nodes such that no other sensor node is inside the circumference of any triangle, as described in the corresponding literature. Delaunay triangulation and the Voronoi diagram are dual graphs. If you have a set of points, you may calculate the Voronoi diagram by first constructing the Delaunay triangulation and then joining the midpoints of the Delaunay edges to produce the Voronoi edges. The following steps are necessary to understand the suggested contribution using these characteristics:

- While deploying sensor nodes, the suggested method uses a random placement of nodes.
- The next step is to use the Delaunay Triangulation method to find gaps in the coverage.
- In the last stage, Coverage hole correction, the proposed approach moves the auxiliary sensor node using the IAOA algorithm and provides the solution for restoring coverage.

The rest of the work is structured like this: issue formulation is outlined, then the recommended technique is shown in detail. Then comes the interpretation of the results, followed by visual representations of the data.

Problem Formulation

Small, low-power wireless devices that can collect and relay information about their surroundings form what are called Wireless Sensor Networks (WSN). Wireless sensor networks (WSNs) have several uses, including in the fields of environmental monitoring, healthcare, and security, to name a few. Making sure the network is well-covered so that data may be sent and received without interruption is a major difficulty in WSN. Yet, there might be blind spots in the network in places where sensors can't reach. These voids in coverage may have a major impact on the efficiency of the network, perhaps leading to dropped connections, lost data, or a diminished overall capacity. Thus, it is crucial to maintain network dependability and efficiency that coverage gaps in WSNs be identified and remedied. Considering that sensor nodes in WSN are deployed at random. Coverage hole repair is the only issue addressed here.

Mitigation of Coverage Hole

It might be difficult to determine the best way to close any gaps in coverage that have been uncovered. Selecting the optimal spot to deploy additional sensor nodes, tweaking the transmission power of existing nodes, and other optimisation strategies are all examples of ways to increase the network's coverage area. For updating the locations of particles or persons in the search space, the arithmetic optimisation algorithm (AOA) uses elementary arithmetic operations such addition, subtraction, multiplication, and division. The fast convergence and low complexity of AOA have made it a popular choice. Nevertheless, similar to other optimisation methods, AOA has the problem of being stuck in local optimums and so failing to adequately explore the search space [15]. Researchers have suggested a number of strategies to enhance AOA's exploratory phase in response to this difficulty. As seen in equation 3 of [13], the suggested technique adds a cognitive element (a) to the classic IAOA's exploratory phase.

$$x_{i,j}(C_{iter}+1) = \begin{cases} a \times \left(best(x_j) \div MOP + \epsilon\right) \times ((UB_j - LB_j) \times \mu + LB_j)) + (1-a) \times (best(x_j) - local(j)), & if\ r2 < 0.5 \\ best(x_j) \times \left((UB_j - LB_j) \times \mu + LB_j\right) \times MOP, & else \end{cases} \quad (2.1)$$

Proposed Methodology

The recommended technique may be seen as new since it involves first choosing a randomly selected set of sensor nodes to deploy across the target WSN region, and then using the Delaunay Triangulation method to identify the existence and location of coverage gaps. Thus, a new subset of supplemental nodes is constructed and provided as an input to the optimizer, which subsequently generates a first set of random solutions. The fundamental cardinalities of the IAOA technique may be summarised as follows: Throughout the process, three sequential objective functions are computed, each of which influences the behaviour of the other. This procedure's novel contribution is directed squarely at the Exploitation stage of the IAOA, when a new cognitive element is added. This function, upon convergence, generates a corresponding logistic map that is provided into the updating procedure as an input. The outcome of this process is the best possible locations for placing sensors. Figure 1 provides a high-level overview of the suggested approach, which proposes using IAOA in detecting the WSN non-triangular coverage gaps by means of the Voronoi diagram implementation and the Delaunay Triangulation methods. There are three main steps to this process: the first phase, which involves the haphazard placement of sensors, the middle phase, which involves the identification of coverage gaps, and the final phase, which involves filling up the gaps that have been identified.

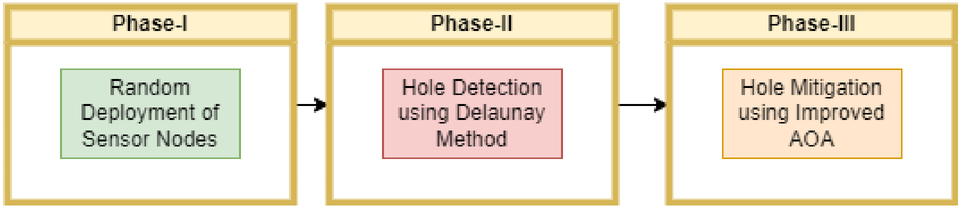

Figure 1 Proposed Methodology

The following sections elaborate on this procedure.

Coverage Hole Mitigation

The Vd (redundant nodes) are deployed during the Exploration phase of IAOA, and holes are patched using the Delaunay technique, as seen in Fig. 2. Mitigating coverage gaps is stated as follows:

$$argmin(w_1 \times N_{hole} + w_2 \times \sum_{i\in(1,N)} A_{hole,i})$$

Weights w1 and w2 are chosen at random here (0,1). The three objective functions are denoted by the symbols f1, f2, and f3 in Equation 3.3.

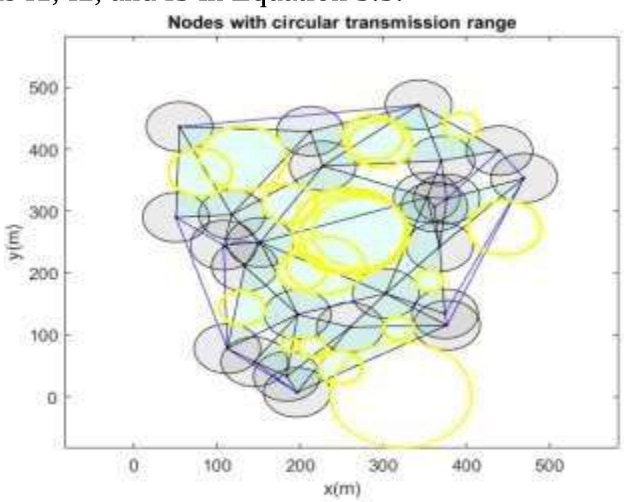

Figure 2 Coverage Holes Detected through DT

The ideal placement of the auxiliary nodes is based on the coverage hole area and the total number of holes. As soon as optimisation begins, a random pool of solutions is established using IAOA, and this serves as the foundation for the hole mitigation design (population). D, M, S, and A estimate where the near-optimal solution may be located along the recurrence trajectory. After the optimal answer has been found, the other answers revert to their initial states. Linearly increasing the MOA function's value from 0.2 to 0.9 prioritises discovery and development. Candidates try to shift away from the near-optimal solution when r1 > MOA, and towards it when r1 MOA.

The current optimum solution is represented as the best candidate solution in each iteration so far, and each AOA optimisation begins with a random set of candidate solutions (X) as shown in the matrix (3.1).

$$X = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,j} & \cdots & x_{1,n} \\ x_{2,1} & \cdots & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n-1,1} & \cdots & \cdots & x_{n-1,j} & \cdots & x_{n-1,n} \\ x_{n,1} & \cdots & \cdots & x_{n,j} & \cdots & x_{n,n} \end{bmatrix} \qquad \text{(matrix 3.1)}$$

Before beginning its operations, the IAOA must choose the search phase (exploration or exploitation). The MOA function, a coefficient calculated by Eq. (3.4), is then utilised in all future iterations of the search.

$$MOA\left(C_{Iter}\right) = Mi + C_{Iter} \times \left(\frac{Max - Min}{M_{Iter}}\right) \qquad (3.4)$$

By plugging MO(Citer) into Eq. (1), we may determine the value of the function after t iterations (3.4). Citer, the current iteration, may be between 1 and Miter, the maximum number of iterations. Minimum (previous) and maximum (current) values of the accelerated function are represented by Min (former) and Max (latter).

In order to discover a better solution, IAOA's exploration operators employ one of two basic search techniques (the Division (D) search strategy or the Multiplication (M) search strategy, both of which are modelled in Eq. 2.1) to randomly explore the search region across several places. If r1 > MOA, then D or E will be used in the following step of searching, an exploratory search, initiated by the Math Optimizer accelerated (MOA) function.

For any arbitrary r1, M is run. The operators used in this research tend to converge in that direction. At this point, we have one operator (D) restricted by r2 0.5 and another operator (M) ignored till D finishes its current task. In the event that D is unable to carry out the requested task, the backup operator M will be contacted (r2 is another random number). A stochastic scaling coefficient is considered for the component in order to build more varied courses and explore a larger range of the search space. We utilised the simplest rule possible to simulate the behaviour of arithmetic operators. The suggested technique improves upon the performance of conventional IAoA with non-linear operational assessment by including a cognitive element (a) into the exploration phase (from equation 3 in [13]). The following equations are proposed for use in the explorer parts of this research as location update equations.

Operators in IAoA's exploration phase randomly probe different parts of the map, using two different kinds of searches (the division D and multiplication M methods) in the hope of finding a more optimal answer. Notations include best(xj), which indicates the jth position in the best-obtained solution so far, xi,j(CIter + 1), which indicates the ith solution in the next iteration, and xi,j(CIter), which indicates the ith solution at the jth location in the current iteration. Upper and lower limits of the jth position are denoted by UBj and LBj, where c is a tiny integral number. The search parameters may be adjusted with the help of the variable.

$$MOP(C_{Iter}) = 1 - \left(\frac{C_{Iter}}{M_{Iter}}\right)^{1/\alpha} \qquad (3.5)$$

where MOP is a coefficient, t is the current iteration, MOP(CIter) is the value of the function at the tth iteration, CIter is the current iteration, and (MIter) is the maximum number of iterations. is a delicate characteristic that controls how well exploited data is over time.

The exploitation phase of searching entails exploitation search by executing S or A if the value of the MOA function for the condition of r1 is not larger than the current MO(CIter). Exploitation operators of IAOA mimic the Subtraction (S) search strategy and the Addition (A) search strategy in Eq. (3.6) to find a better solution if one exists.

$$X_{i,j}(C_{Iter}+1) = \begin{cases} best(x_j) - (MOP) \times \left((UB_j - LB_j) \times \mu + LB_j\right), & r_3 < 0.5 \\ best(x_j) + (MOP) \times \left((UB_j - LB_j) \times \mu + LB_j\right), & otherwise \end{cases} \qquad (3.6)$$

As is evident in Fig. 3, this stage makes efficient use of the search area by conducting a thorough search, as the first operator (S) is confined by r3 0.5 and the other operator (A) will be ignored until the first operator has finished its current task . If S is unable to complete the task, the other operator (A) will be tasked with it. Comparing the phase-one partitioning to the phase-two operations is straightforward. Knowing this, those doing exploitative searches (S and A) avoid focusing just on the nearby area. Exploratory search strategies benefit from this strategy since it helps them identify the best answer while protecting a wide range of options. The parameters are set up to produce a random number at each iteration, keeping the excitement of discovery going strong from the very first to the very last. The IAOA algorithm ends when the last criterion is satisfied. Fig. 3 is a flowchart that outlines the complete proposed technique, and Algorithm 2 is pseudo-code that describes the coverage hole mitigation procedure.

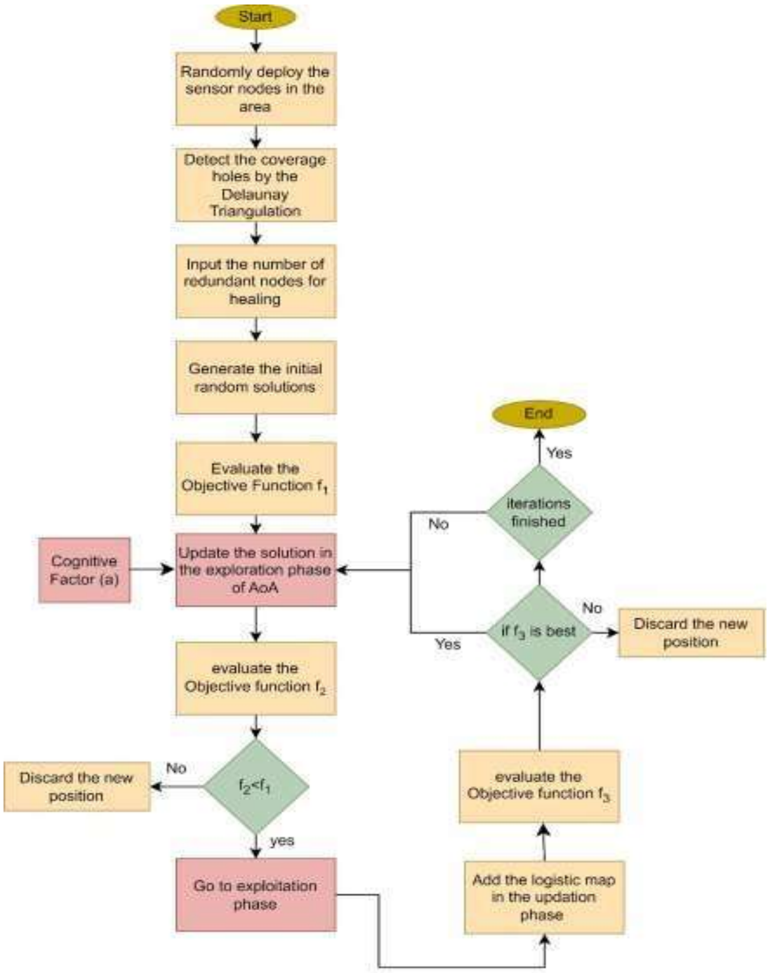

Figure 3 Flowchart of WSN hole detection using IAOA

Algorithm 2 Pseudo Code for Coverage Hole Mitigation

Input: V_n is the number of sensor nodes in the WSN area, V_d is the number of redundant nodes for healing, w_1 and w_2 be the weights for the objective function, and $max\_iterations$ be the maximum number of iterations.

1. Randomly deploy V_n sensor nodes in the WSN area.
2. Detect coverage holes using the Delaunay triangulation method.
3. Input the V_d for healing.
4. Generate the initial random solutions.
5. Evaluate the objective function $f_1 = w_1 \times N_1 + w_2 \times A_1$ for the current positions of sensor nodes.
6. **Repeat** Steps 1 to 5 until the maximum number of iterations is reached:
   1: Update the solution in Improved Exploration Phase using the IAOA algorithm.
   2: Evaluate the objective function $f_2 = w_1 \times N_1 + w_2 \times A_1$ for the new positions generated in Step 7.
   3: **If** $f_2 > f_1$, **then**
      a. performs the Exploitation phase using IAOA algorithm.
   4: **else**
      a. discards the new position.
   5: **end**
   6: Add the logistic map in the updating phase.
   7: Evaluate the objective function $f_3 = w_1 \times N_1 + w_2 \times A_1$ for the updated positions generated in Step 3.
   8: **If** f_3 is better than f_1 **then**
      a. **Set** V_n to the updated positions $f_1 = f_3$.
   9. **Else**
      a. discards the new position.
7. If the maximum number of iterations is reached, end the process, else repeat Step 6.

Output: The final positions of sensor nodes

To sum up, the process begins with a set of n randomly selected sensor nodes Vn deployed over the target WSN area, followed by the implementation of the Delaunay Triangulation method to detect the presence and position of the corresponding coverage holes, and finally the generation of a new subset of supplementary nodes Vd. The optimizer then takes Vd as an input and generates a random set of solutions to start the process. As shown in Algorithm 2, the IAOA approach kicks off with the calculation of the objective function f1. In the meanwhile, f2 is computed as an objective function, and if it is larger than f1, the Exploitation phase of the IAOA is initiated, else the associated position value is deleted. When convergence is reached at this stage, the newly created logistic map is used as a starting point for the revision procedure. Finally, we compute the final objective function f3, and if it's better than f1, we add the new locations to the Vn and set f1 = f3. If not, we throw out the new and old places. The outcome of this process is the best possible locations for placing sensors.

Results Evaluation

Simulation Environment

A specialised MATLAB implementation conducts simulations on a 500 x 500 metre grid with 25 nodes, assuming an omnidirectional antenna for all nodes, to test the feasibility of the proposed technique. Each node may serve as a signal transmitter or receiver within a range of 40 metres. The recommended method accounted for the three spare nodes needed to patch the coverage gap.

Evaluation and the State of Art Comparison

A two-ray ground wave propagation model and a time step of 150 ms were used to create this simulation. Various evaluations employed a large variety of redundant node counts. The number of holes and the total hole area were used to quantify the primary results. The outcomes are evaluated against those of the most cutting-edge AOA optimisation methods now in use. Redundant nodes wait in the background until the WSN's control node sends them the appropriate information to activate and fill up a coverage hole. Transmission of control/active signals to nodes and their algorithmic implications were mostly outside the scope of this research. The VdS are released into the wild on the assumption that they have received the active status signal and are capable of moving using the random mobility model. Figure 6 compares the convergence curves obtained using the AOA and the proposed method IAOA for finding the optimal value of d=3. Both the (AOA) and (IAOA) optimisation strategies take into account the presence of the three spare nodes.

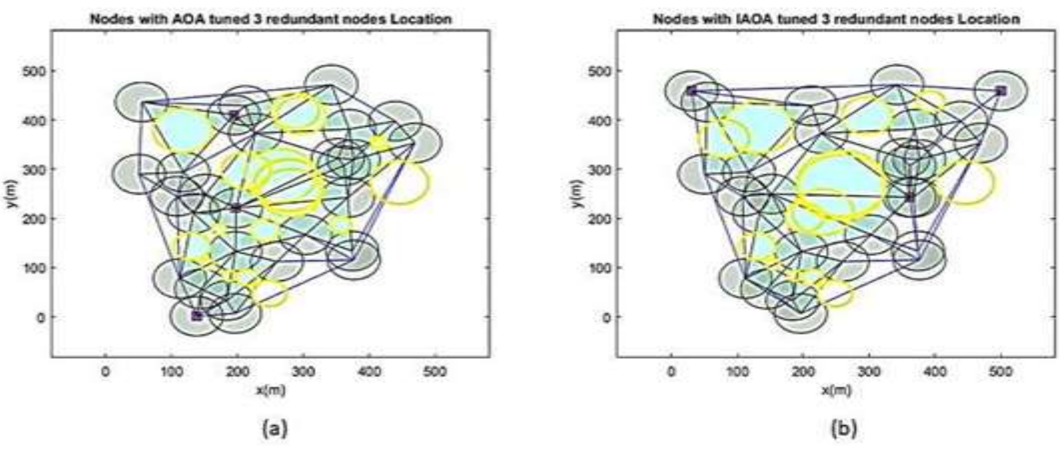

Figure 4 Coverage Hole Healing through (a) AOA (b) IAOA

Figure 4 provides a concise illustration of the statistics acquired with regard to the number of holes found and their corresponding areas, which are among the aggregated performance indicators. In both of our optimization plans, we have accounted for three spare nodes.
After positioning redundant nodes, the cyan circles in Fig. 4 depict the appropriate circular hole areas. In the diagram, the empty squares show nodes that were not utilized. The conventional AOA tends to terminate sooner than the suggested IAOA, making it relatively more difficult to find the optimal placement of superfluous nodes, which is why the holes in 4(a) have a larger total area than those in 4(b). As can be seen in Fig. 4(b), the coverage holes are less as a consequence of the IAOA optimization thanks to the novel cognitive enhancement used in the procedure. The area and number of holes in the covering have been reduced thanks to the proposed strategy, as illustrated in Fig. 5. (b). Furthermore, we see that the overall number of holes drops as coverage grows, but the ratio of holes to covered area has not changed.

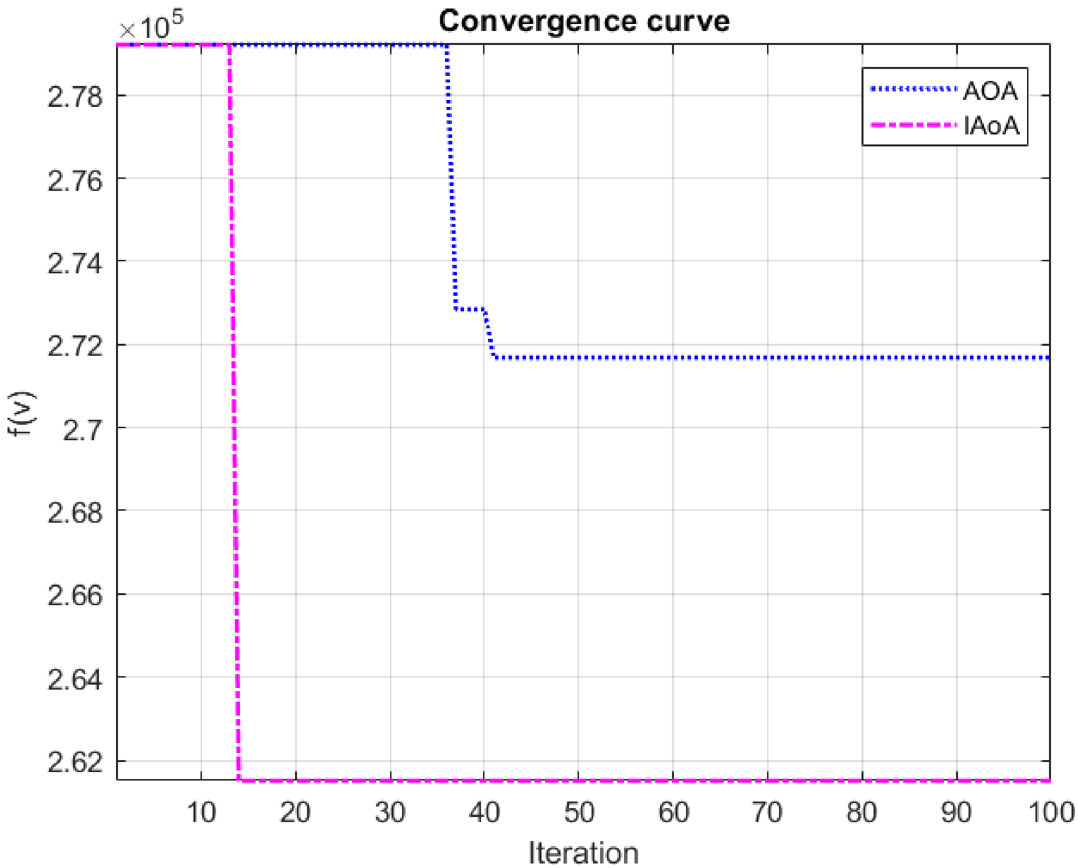

Figure 5 Comparison of Convergence Curve obtained for AOA and IAOA

A more efficient methodology may be identified as having a smaller convergence curve when comparing different optimization methods. The two convergence curves are compared in Fig. 5. If the saturation value can be attained more quickly, then the strategy is more ideal. Both minimizing the overall hole area and minimizing the total number of holes are top priorities for us. On the ordinate of Fig. 5 are shown the objective function's measurements. With the proposed method, convergence occurred by the thirteenth iteration, while further iterations may be necessary for alternative approaches. The upgraded IAOA optimization's exploration phase sped up the convergence time. After using the proposed strategy for 10 iterations, we discovered a faster rate of convergence compared to the original. After 100 cycles, our technique converges to a lesser number than AOA did, but it outperformed AOA while patching coverage gaps because it included redundant nodes that are still active in the network. It's possible that taking part in the conversation has already drained them considerably. Table 1 shows that compared to state-of-the-art approaches like PSOCGSA, which can only cover up to the values of 6.87% of the hole area, our proposed strategy has the ability to cover up to 59.15% of the hole area. The simulation was run with 100 iterations per optimization, and the results are shown visually.

Table 1 State-of-art Comparison

	IAOA	PSOCGSA
Percentage of Hole Coverage Area after healing	40.85%	93.13%
Number of Redundant Nodes	3	3

Conclusion

The capacity to do comprehensive, detailed monitoring is a major benefit of today's large-scale mobile networks and the wireless sensors they include. Yet, in a WSN, Coverage Holes and substantial overlap might arise from the random initial placement of sensor nodes, leading to reliability and quality of service issues. The coverage gaps problem in WSN can only be addressed by making efficient use of existing resources. Without a remedy, coverage gaps in WSN are only anticipated to become worse, and there is presently no way to eradicate them. This implies that a completely functional and efficient sensor network requires the identification and closure of coverage gaps. Coverage gaps will always exist, therefore it's crucial to have a plan in place that can quickly identify and fix them. Nevertheless, only the most fundamental tools are available in a real-world test environment, making it difficult to identify the cause of a coverage gap. The proposed method has the potential to improve coverage hole mitigation by as much as 59.15%, or 10.41% more than the state-of-the-art PSOCGSA.

References

1. Mehta, S., & Malik, A. (2020). A swarm intelligence-based coverage hole healing approach for wireless sensor networks. EAI Endorsed Transactions on Scalable Information Systems, 7(26), e8-e8.
2. Shivalingegowda, C., & Jayasree, P. V. Y. (2021). Hybrid gravitational search algorithm-based model for optimizing coverage and connectivity in wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing, 12, 2835-2848.
3. Tsai CW, Tsai PW, Pan JS, Chao HC (2015) Metaheuristics for the deployment problem of WSN: a review. Microprocess Microsyst 39:1305–1317
4. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS’95. Pro-ceedings of the Sixth International Symposium on Micro Machine and Human Science. IEEE, pp 39–43

5. Karaboga D (2005) An idea based on honeybee swarm for numerical optimization. Tech Rep TR06 Erciyes Univ 200:1–10
6. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17:4831–4845
7. Wang GG, Gandomi AH, Alavi AH (2014) Stud krill herd algorithm. Neurocomputing 128:363–370
8. Wang GG, Gandomi AH, Alavi AH (2014) An effective krill herd algorithm with migration operator in biogeography-based optimization. Appl Math Model 38:2454–2462
9. Wang GG, Guo L, Gandomi AH, Hao GS, Wang H (2014) Chaotic Krill Herd algorithm. Inf Sci 274:17–34
10. Wang G, Guo L, Wang H, Duan H, Liu L, Li J (2014) Incorporating mutation scheme into krill herd algorithm for global numerical optimization. Neural Comput Appl 24:853–871
11. Wang GG, Deb S, Gandomi AH, Alavi AH (2016) Opposition-based krill herd algorithm with Cauchy mutation and position clamping. Neurocomputing 177:147–157
12. Wang H, Yi JH (2018) An improved optimization method based on krill herd and artificial bee colony with information exchange. Memetic Comput 10:177–198
13. Rizk-Allah RM, El-Sehiemy RA, Wang GG (2018) A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution. Appl Soft Comput J 63:206–222
14. Selvaraj, A., Patan, R., Gandomi, A. H., Deverajan, G. G., & Pushparaj, M. (2019). Optimal virtual machine selection for anomaly detection using a swarm intelligence approach. Applied Soft Computing, 84, 105686.