Indian Journal of Pure & Applied Physics
Vol. 63, September 2025, pp. 847-858
DOI: 10.56042/ijpap.v63i9.20363

Dynamic Deployment of Mobile Sensor Nodes to Achieve Target Coverage

Himansu Bhushan Mohapatra* & Mrutyunjay Rout

Department of Electronics and Communication Engineering, National Institute of Technology, Jamshedpur 831 001, India

Received: 5th June 2025; accepted: 14th July 2025

Target coverage is an essential function in wireless sensor networks (WSNs), especially in applications like surveillance, environmental monitoring, and disaster response. When mobile sensor nodes are randomly deployed, achieving efficient and uniform target coverage is challenging due to limited mobility, energy constraints, and uneven distribution. This study presents a hybrid optimization approach that integrates the Virtual Force approach (VFA) with Particle Swarm Optimization (PSO) to dynamically position mobile sensor nodes for optimal target coverage with the objective of maximizing coverage and minimizing movement. The VFA is used to benefit from local interactions between nodes and guide early movement using attractive and repulsive virtual forces, facilitating rapid dispersion and preliminary coverage. We then use Particle Swarm Optimization (PSO) to globally optimize node placements, enhancing coverage quality and reducing redundant overlaps and movement expenses. The effectiveness of the proposed VFA+PSO hybrid algorithm is assessed based on two primary metrics: coverage quality, which is defined by the number of sensors observing each target, and convergence rate, demonstrated through the iterations necessary to achieve a steady state. Results from the simulation validate the effectiveness of the proposed method using different deployment scenarios of sensors by achieving better coverage more quickly and with less movement of sensors, making it perfect for changing wireless sensor network applications that require flexible and energy-saving deployment strategies.

Keyword: Dynamic deployment, Target coverage, Hybrid algorithm, VFA, PSO, Coverage quality, Convergence rate

1 Introduction

Wireless Sensor Networks (WSNs) have become an effective tool for data collecting and monitoring in various applications, from industrial automation and disaster management to military surveillance and environmental sensing. Monitoring a set of predetermined points or areas of interest is a crucial operational need for target coverage in many of these applications. Static placement of sensor nodes, however, often results in inadequate coverage in real-world deployments because of barriers, random node distribution, and a lack of previous environmental information.

Mobile sensor nodes have emerged as a flexible and adaptable approach to address these difficulties. In comparison with static nodes, mobility sensors may relocate themselves after deployment to optimize coverage, enhance connection, and prolong network durability. Dynamic deployment of mobile sensors is challenging, particularly in finding movement strategies that balance coverage quality, energy efficiency, and deployment time.

*Corresponding author: E-mail: hbmiitkgp@gmail.com

Conventional deployment techniques, such as random walks or greedy algorithms, often fail to provide efficient and scalable coverage. Optimization-based techniques have been considered to reduce these limitations. Virtual Force Algorithms (VFA) and Particle Swarm Optimization (PSO) have shown promising results. VFA uses virtual force fields to direct sensor movement according to local interactions, while PSO applies a population-based search, inspired by social behaviour, to identify global optima. Each method has distinct limitations: VFA may converge rapidly but has chances of trapping in local optima, whereas PSO provides global search capabilities but may need more time for convergence and refinement of node placements.

It presents a hybrid VFA+PSO algorithm that integrates the advantages of both methodologies for the dynamic deployment of mobile sensor nodes to ensure optimal target coverage. The suggested methodology employs VFA to create an initial feasible deployment through local repulsion and attraction forces, then implements PSO to enhance the locations by optimizing a global objective function with the objective of maximizing coverage and minimizing movement. The hybrid technique facilitates rapid initial distribution, followed by meticulous optimization, resulting in enhanced coverage performance with fewer iterations.

To evaluate the performance of the proposed strategy, we provide two main metrics, coverage quality, defined as the quantity of sensor nodes successfully monitoring each target and convergence rate, quantified by the number of iterations necessary to achieve a stable deployment state. The simulation findings indicate that the hybrid VFA+PSO algorithm substantially surpasses both individual approaches and baseline strategies across all parameters, providing an efficient solution for dynamic sensor deployment in target-oriented wireless sensor network settings.

2 Related Work

The target coverage problem in wireless sensor networks (WSNs) has been widely investigated due to its fundamental role in ensuring reliable and efficient monitoring of specific points or regions of interest. Various strategies have been proposed for optimizing the deployment of sensor nodes to maximize coverage while maintaining energy efficiency, connectivity, and scalability.

Yarinezhad and Hashemi[1] introduced a deployment approach focusing on reducing coverage redundancy while improving target monitoring efficiency. Their method demonstrated performance improvements under fixed deployment scenarios but lacked adaptability for dynamic environments. Mini *et al.*[2] proposed a sensor deployment and scheduling framework that addressed energy conservation through activation scheduling; however, their method primarily focused on static nodes and did not handle post-deployment mobility.

Cardei *et al.*[3] and Zou & Chakrabarty[4] laid foundational work on energy-efficient target coverage and localization strategies. Their studies introduced the use of cover sets and probabilistic deployment models, respectively, which provided important information about balancing coverage and energy usage. Akram *et al.*[5] advanced the concept by incorporating adaptive sensors with intelligent decision-making to enhance target coverage dynamically, although their algorithmic complexity remained high.

Random deployment challenges were addressed by Katti[6], who utilized cover sets in stochastic environments, and Liang *et al.*[7], who focused on maximizing coverage with mobile nodes. While these works advanced coverage potential, they lacked integration of cooperative optimization strategies for efficient movement control. Singh *et al.*[8] highlighted the importance of quality of service (QoS) in target coverage, introducing performance metrics relevant to real-world deployment scenarios.

Njoya *et al.*[9] and Banoth *et al.*[10] proposed hybrid schemes that considered both coverage and connectivity, achieving balanced network performance. Arivudainambi *et al.*[11] introduced metaheuristic approaches such as the Cuckoo Search Algorithm for joint optimization of sensing range and scheduling, but scalability and convergence remained concerns. Similarly, Manju and Kumar[12] investigated the K-coverage problem to ensure fault tolerance, while Mottaki *et al.*[13] tackled Q-coverage in over- and under-provisioned networks using genetic algorithms.

Cheng and Wang[14] contributed to barrier coverage optimization, highlighting challenges in linear target coverage scenarios. Finally, the cooperative particle swarm optimization approach by Van den Bergh and Engelbrecht[15] provided a basis for distributed and cooperative control strategies in dynamic optimization problems.

Binh *et al.*[16] aimed to maximize target coverage and extend network lifetime in mobile WSNs. They proposed a heuristic node placement strategy based on evolutionary intelligence. The results showed improved coverage and energy efficiency. However, the approach may struggle in highly dynamic environments due to limited adaptability. Muhammad and Nam[17] focused on optimizing sensor deployment for continuous area coverage. Using an optimization framework, they achieved reliable, long-term monitoring. The main limitation lies in the assumption of static environmental conditions, which may not hold in real-world deployments. Table 1 presents a summary of the related work, highlighting the objectives, contributions, and limitations.

The above studies have made significant progress in advancing deployment strategies for wireless sensor networks. However, a common limitation remains, that many approaches either focus solely on static deployments or lack effective integration of local mobility control with global

Table 1 — Summary of Related Works

Ref.	Authors	Objective	Classification	Contributions	Limitations
[1]	Yarinezhad & Hashemi	Deployment strategy to reduce coverage redundancy	Fixed deployment	Improved monitoring efficiency by minimizing redundancy	Lacks adaptability in dynamic environments
[2]	Mini et al.	Sensor scheduling for energy conservation	Static node scheduling	Activation scheduling to extend network life	Does not support mobile sensor nodes
[3] [4]	Cardei et al., Zou & Chakrabarty	Energy-efficient coverage using cover sets and probabilistic models	Foundational energy-efficient models	Balanced coverage and energy with foundational strategies	Limited dynamic adaptability
[5]	Akram et al.	Adaptive sensor deployment with intelligent decisions	Adaptive/dynamic model	Enhanced dynamic target coverage	High algorithmic complexity
[6]	Katti	Random deployment using cover sets	Random deployment	Improved target coverage in stochastic environments	No optimization for sensor movement
[7]	Liang et al.	Mobile node deployment to improve coverage	Mobile deployment	Maximized coverage using node mobility	Lacks cooperative optimization
[8]	Singh et al.	QoS-driven performance metrics in WSNs	QoS-based deployment	Introduced real- world performance metrics	General approach, not coverage- optimized
[9], [10]	Njoya et al., Banoth et al.	Hybrid schemes for coverage and connectivity	Hybrid deployment	Balanced network performance	Moderate scalability
[11]	Arivudaina mbi et al.	Metaheuristic (Cuckoo Search) for sensing and scheduling	Metaheuristic optimization	Joint sensing range and scheduling optimization	Scalability and slow convergence
[12]	Manju & Kumar	K-coverage for fault tolerance	Fault-tolerant coverage	Ensured multi-node coverage	Assumes uniform deployment conditions
[13]	Mottaki et al.	Genetic algorithm for Q-coverage	Redundant-aware optimization	Addressed over- and under- provisioning issues	May lead to uneven coverage
[14]	Cheng & Wang	Linear barrier coverage optimization	Barrier coverage	Handled specific linear coverage challenges	Limited to specific scenarios
[15]	Van den Bergh & Engelbrecht	Cooperative PSO for dynamic optimization	Swarm intelligence	Basis for distributed optimization in WSNs	General-purpose, not coverage- specific

optimization techniques. Furthermore, the core performance indicators such as convergence rate and coverage quality have received limited attention, particularly in dynamic and real-time deployment environments.

To bridge this gap, our work proposes a hybrid VFA+PSO algorithm that integrates the local responsiveness of the Virtual Force Algorithm with the global optimization capability of Particle Swarm Optimization. This combination enables dynamic repositioning of mobile sensor nodes, achieving enhanced target coverage, faster convergence, and reduced energy consumption compared to existing approaches.

3 Preliminary and Problem Statement

3.1 Preliminary

Target coverage in a randomly deployed sensor network refers to the network's ability to monitor or observe a set of specific points or areas of interest, called targets, using its sensors. The concept of target coverage is based on the following definition

Target: A fixed location or object in the region of interest that must be monitored.

Coverage: A target is said to be covered if it lies within the sensing range of at least one sensor.

k-Coverage: Each target is covered by at least k different sensors. where k is a predefined integer constant.

Q-Coverage: In Q-coverage, each target T_j must be monitored by at least q_j sensor nodes, where j ranges from 1 to n, and n is the total number of targets.

Connected: A sensor is considered connected if it can communicate with at least one other sensor.

Random Deployment: Sensors are distributed in the field of interest without a deterministic pattern (e.g., dropped from an aircraft), which results in non-uniform coverage.

3.2 Virtual Force Algorithm

The Virtual Force Algorithm (VFA) as a foundational component for optimizing sensor deployment. VFA simulates a physical force-based system in which targets exert attractive forces on sensor nodes, while neighboring sensors exert repulsive forces to maintain connectivity and avoid collisions. Each sensor updates its position by computing the net virtual force acting upon it, leading to an iterative self-organization of the network. The algorithm's distributed and intuitive nature enables rapid convergence, making it suitable for real-time and large-scale deployments. However, due to its gradient-like behavior, VFA is susceptible to getting trapped in local optima, which can result in suboptimal target coverage. The basic mathematical equation used for implementation of VFA is as follows

3.2.1 Attractive Force from Targets

For each target T_j a sensor S_i feels an attractive force (Eq. (1)) if it is not extremely close to that target.

$$\vec{F}_{\text{attract}}^{ij} = \frac{a \cdot (\vec{T}_j - \vec{S}_i)}{|\vec{T}_j - \vec{S}_i|^3} \quad \text{... (1)}$$

Where $\overrightarrow{S_i}$ is the position of sensor i. $\overrightarrow{T_j}$ is the position of target j, a is the attractive force constant. $|\overrightarrow{T_j} - \overrightarrow{S_i}|$ is the Euclidean distance between sensor and target.

The force decreases with distance squared (inverse-square law), but the direction vector is normalized, so effectively the force scales with[1] distance

This encourages sensors to move toward nearby targets more strongly.

3.2.2 Repulsive Force from Other Sensors

Each sensor feels a repulsive force (Eq. (2)) from other sensors within communication range that prevents sensor overlap and excessive clustering.

$$\vec{F}_{\text{repulse}}^{ik} = \frac{r \cdot (\vec{S}_i - \vec{S}_k)}{\|\vec{S}_i - \vec{S}_k\|^3} \quad \text{... (2)}$$

Where r is the repulsive force constant. $\overrightarrow{S_k}$ is the position of another sensor $k \neq i$, The force acts only if $|\overrightarrow{S_i} - \overrightarrow{S_k}| < R_c$, i.e., within communication range.

3.2.3 Total Virtual Force

The net virtual force on a sensor is the vector sum of attractive and repulsive forces Eq. (3)

$$\vec{F}_i = \sum_j \vec{F}_{\text{attract}}^{ij} + \sum_{k \neq i} \vec{F}_{\text{repulse}}^{ik} \quad \text{... (3)}$$

3.2.4 Position Update

Each sensor updates its position using Eq. (4)

$$\vec{S}_i^{\text{new}} = \vec{S}_i + \gamma \cdot \vec{F}_i \quad \text{... (4)}$$

where γ is a scaling factor to control step size.

3.3 Particle Swarm Optimization

To enhance the global search capability of the optimization process, we integrate Particle Swarm Optimization (PSO) into the sensor deployment strategy. In PSO, each candidate solution representing a specific configuration of sensor positions is treated as a "particle" within the search space. These particles iteratively adjust their positions by considering both their own historical best position (personal best) and the best-known position discovered by the swarm (global best). This mechanism allows PSO to effectively explore the solution space and avoid local optima. While PSO offers strong global optimization performance, it typically converges more slowly and may require a well-informed initial population to achieve efficient results. The basic mathematical equation used for implementation of PSO is given in Eqs. (5,6).

$$V_i(t+1) = w(t) * V_i(t) + b_1 * rand * \left(X_{bi}(t) - X_i(t)\right) + b_2 * rand * \left(X_g(t) - X_i(t)\right) \quad \text{... (5)}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad \text{... (6)}$$

Here b_1 and b_2 are the cognitive and social factors and $w(t)$ is the inertia weight.

3.4 Hybridization Strategy

A hybrid VFA+PSO strategy to optimize sensor deployment for effective target coverage and network connectivity. The hybrid method combines the quick, local adjustments of the Virtual Force Algorithm (VFA) with the strong global search abilities of Particle Swarm Optimization (PSO). First, VFA is used to help sensor nodes move towards areas with many targets by creating virtual forces that attract or repel them, resulting in a better starting arrangement focused on coverage. This layout, processed by VFA, is then used to start the particle group in PSO, which fine-tunes the sensor locations by adjusting their speed and position based on the best solutions found personally and globally. The combination allows the system to quickly escape poor local configurations while improving convergence toward a globally optimal deployment. This two-step optimization method combines speed and quality of solutions, leading to better coverage and connectivity than using VFA or PSO by themselves.

3.5 Problem Statement

Let us consider $T = \{t_1, t_2, \ldots, t_n\}$ be a set of n targets, $S = \{s_1, s_2, \ldots, s_m\}$ be a set of m sensor nodes having sensing radius R_S and Communication range R_c

Then a target $t_i \in T$ is considered covered if there exists at least one sensor $s_j \in S$ such that:

$$d\ (t_i, s_j) \leq R_s$$

\i.e. Distance between targets and sensor nodes are less than or equal to sensing radius of sensors. The main objective of target coverage is to maximize the number of covered targets that covers all targets with minimum moving distance while maintaining the connectivity among them. The mathematical expression for the objective function is as follows

3.5.1 Coverage Calculation

Each target t_i is considered covered if any sensor is within its sensing radius R_s.i.e

$$\mathrm{C}(i) = \begin{cases} 1, & \text{if } \exists j \in \{1, \ldots, S\} \text{ such that } \|\mathrm{s}_j - \mathrm{t}_i\| \leq R_s \\ 0, & \text{otherwise} \end{cases}$$

s_j is the position of sensor j and t_i is the position of target i

So, Coverage ratio is calculated using Eq. (7)

$$C_v = \frac{1}{T}\sum_{i=1}^{T} \mathrm{C}(i) \qquad \text{... (7)}$$

3.5.2 Movement Calculation

Compute total movement from initial position to current position for all sensors:

$$M_T(\text{Movement Total}) = \sum_{j=1}^{S} \left\|\mathrm{s}_j^{\text{current}} - \mathrm{s}_j^{\text{initial}}\right\|$$

Then the total movement is normalized by number of sensors (S) using the Eq. (8)

$$M_{Avg}(MovementAverage) = \frac{M_T}{S} \qquad \ldots (8)$$

3.5.3 Connectivity Calculation

A sensor network is connected if there exists a communication path between all pairs of sensors. Two sensors are connected (Eq. (9)) if they are within the communication radius R_c.

$$C_{ij} = \begin{cases} 1, & \text{if } \|\mathbf{s}_i - \mathbf{s}_j\| \leq R_c \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases} \qquad \ldots (9)$$

Then a Breadth-First Search (BFS) is performed starting from any sensor node. If all nodes are reachable, the network is connected i.e.

$$Connected = \begin{cases} true, if\ all\ nodes\ visited\ in\ BFS \\ \qquad false, otherwise \end{cases}$$

3.5.4 Final Fitness Function

The fitness is a weighted function combining the maximize coverage, minimize movement and ensuring connectivity given in Eq. (10).

$$F(FitnessFunction) = \alpha.C_v + \beta.M_{Avg} \qquad \text{... (10)}$$

4 Simulation of Proposed Algorithm

The primary objective is to ensure target coverage with minimal relocation of sensors, while maintaining connectivity across the sensing network. In our simulations, sensor nodes are randomly deployed within the region of interest. We use MATLAB to implement and run multiple scenarios to evaluate the performance of the proposed algorithm. To evaluate the performance of the proposed algorithm, two key metrics are considered, coverage quality and convergence rate (or deployment speed). Coverage quality refers to the number of sensor nodes monitoring each target, while convergence rate denotes the number of iterations needed for the algorithm to reach a steady state. We assume that the network deployment area spans from 100m to 100m, and we need to cover 20 target nodes. The simulation generates target locations at random and saves them into an array for further processing. We vary the number of sensor nodes between 10 and 30. The sensing range and communication range of each sensor node are 10 m and 20 m, respectively. Simulation parameters are summarized in Table 2. We report results based on an average of 25 runs in each configuration. To execute our proposed algorithms in PSO, we considered an initial population of 30 particles and set the values of parameters b1, b2 and w to 2, 2, and 0.6, respectively. Similarly, in VFA, we considered the attractive force coefficient, repulsive force coefficient, and Ɛ as 1.5, 1.0, and

0.001, respectively. We use various algorithms, including Algorithms 1, 2, and 3, to implement the proposed algorithm and compare the results.

Table 2 — Simulation Parameters

Parameter	Value
Field Size	100m x 100m
No. of Sensor Nodes	10-30
Sensing Radius (R_s)	10-15m
Communication Range (R_c)	20-30m
Number of Particle	30
Number of Sensor	20 to 30
Number of Target	5 to 20
Attractive Force Coefficient (a)	1.5
Repulsive Force Coefficient (r)	1.0
α	1
β	0.01
γ	0.5
w	0.6
b_1, b_2	2,2
Iterations	25-100

Algorithm 1: VFA for Sensor Node Repositioning

Input:
1 Initial positions of sensor nodes, Positions of targets, Sensing range of sensor nodes, Size of the deployment area, Number of iterations
Output:
2 Final sensor positions after applying VFA.
Begin:
3 Initialize algorithm parameters.
4 For each iteration
5 For each sensor node i
6 For each target j
7 Compute distance: d = (sensor_i − target_j).
8 If d > ε then:
9 Compute direction: dir = (target_j − sensor_i) / d.
10 Compute attractive force: $\vec{F}^{ij}$
End For
11 For each sensor node k ≠ i
12 Compute distance: d = (sensor_i − sensor_k).
13 If ε < d < Rs, then:
14 Compute direction: dir = (sensor_i − sensor_k) / d.
15 Compute repulsive force: $\vec{F}^{ik}$
End For
16 Compute Total Force: $\vec{F}_i$
17 If Total Force: $\vec{F}_i > 1$, then:
18 Normalize total force
19 Update sensor position: $\vec{S}^{new} = \vec{S}_i + \alpha \cdot \vec{F}_i$
End For End For
20 Return Final sensor positions.

Algorithm 2: PSO for Sensor Node Repositioning

1 Initialize algorithm parameters
2 Initialize random particle position and velocity with the ROI
3 for each particle
4 Calculate the fitness function:F
5 if (F(Xi) < F(Xbi))
6 then Xbi = Xi
7 *end*
8 end
9 Evaluate global best: Xg=min of { Xbi }
10 for each particle
11 Update Velocity
12 Update Position
13 *end*
14 Repeat until the stopping condition is not reached
15 end

Algorithm 3: Proposed Hybrid Algorithm (VFA+PSO)

Initialization
1 Random deployment of sensors in the ROI
2 Initialize algorithm parameters
VFA Phase
3 Repeat for VFA iterations For each sensor
Calculate the virtual attractive force from uncovered barrier points Calculate repulsive forces from nearby sensor positions
Compute total force Update sensor position

PSO Phase
4 Initialize particles' velocities, personal and global bests
5 Repeat for PSO iterations For each sensor(particle)
Evaluate fitness of position using the defined fitness function. Update personal best and global best
Update velocity using velocity equation Update position using position equation
6 Return global best position as the final sensor position
7 Repeat until the stopping condition is not reached

5 Results

The results from simulations using the proposed hybrid algorithm with Virtual Force Algorithm (VFA) and Particle Swarm Optimization (PSO) for placing sensors in wireless sensor networks. Figures 1-7 illustrate the performance of the algorithm under various initial sensor deployment conditions. Each figure comprises four subfigures: (a) initial sensor positions, (b) final sensor positions with movement paths, (c) average moving distance versus iteration number, and (d) number of monitoring sensors versus iteration number.

In all the figures, subfigure (a) displays various starting positions of the sensors, while subfigure (b) shows how the sensors move to new locations using the hybrid optimization algorithm. The movement paths in subfigure (b) indicate that sensors efficiently reposition themselves toward optimal locations, improving target coverage and minimizing redundancy. Subfigures (c) and (d) further validate the algorithm's convergence behavior. The average moving distance (subfigure c) generally exhibits a rapid decline and stabilizes within approximately 20 to 30 iterations, indicating efficient convergence. The number of monitoring sensors (subfigure d) increases over iterations and eventually plateaus, showing that the network reaches a stable and effective coverage state.

The analysis of the seven deployment scenarios from Scenario 1 to 7, demonstrates (Table 3) the robustness and efficiency of the proposed hybrid VFA+PSO algorithm. The coverage quality across all scenarios remains consistently high, ranging from 92% to 97%, with the best performance observed in the centre deployment scenario (97%). The convergence rate, measured by the number of iterations required to reach a steady state, varies between 20 and 30 iterations. The fastest convergence occurs in the centre deployment scenario (20 iterations), while corner-biased and clustered scenarios require slightly more iterations (up to 30) due to higher initial unevenness. The number of monitoring sensors at convergence remains close to optimal in all cases, with 46 to 49 out of 50 sensors effectively contributing to target monitoring. These results confirm that the algorithm adapts well to diverse and challenging initial configurations, achieving optimal coverage with efficient sensor utilization and rapid convergence.

Scenario 1: Random Deployment

In this scenario (Fig. 1), sensors are initially dispersed randomly across the field. The algorithm successfully relocates sensors to cover the target area uniformly. About 25 iterations achieve convergence, and nearly all sensors contribute to monitoring, indicating effective utilization and optimal deployment.

Scenario 2: Bottom-Left Clustered Deployment

Here (Fig. 2), sensors begin clustered in the bottom-left quadrant. The algorithm demonstrates significant repositioning capabilities, as shown by the extended movement paths. Despite the challenging initial distribution, the average movement stabilizes efficiently, and the monitoring sensor count rises steadily, confirming the algorithm's adaptability.

Scenario 3: Bottom-Right Clustered Deployment

This setup is similar to Fig. 3 but with sensors initially placed in the bottom-right corner. The relocation demand is similarly high, and convergence occurs slightly later compared to Figs. 1- 2. Nonetheless, the final sensor distribution ensures robust target monitoring, validating the approach's reliability.

Scenario 4: Centre Deployment

Sensors start near the center of the monitoring area (Fig. 4), offering an inherently balanced deployment. Consequently, sensor movement is minimal, and

Table 3 — Comparison of Deployment Scenario

Scenario	Coverage Quality (%)	Convergence Rate (Iterations)	Number of Monitoring Sensors
1. Random Deployment	Uniform coverage achieved across the field (95%)	Converges in 25 iterations.	48 sensors contribute to monitoring out of 50.
2. Bottom-Left Clustered	Effective redistribution, good coverage despite initial unevenness (93%)	Converges in 28 iterations.	47 sensors contribute to monitoring out of 50.
3. Bottom- Right Clustered	Robust target monitoring after significant relocation (92%)	Converges in 30 iterations.	46 sensors contribute to monitoring out of 50.
4. Centre Deployment	High-quality, balanced coverage with minimal repositioning (97%)	Fastest convergence due to favourable start. Converges in 20 iterations.	49 sensors contribute to monitoring out of 50.
5. Top-Left Clustered	Optimal coverage despite corner bias (94%)	Quick stabilization after initial high movement. Converges in 27 iterations.	47 sensors contribute to monitoring out of 50.

convergence is faster. The average moving distance is the lowest among all scenarios, and the number of monitoring sensors increases rapidly, highlighting the algorithm's efficiency under favorable initial conditions.

Scenario 5: Top-Left Clustered Deployment

In this scenario (Fig. 5), sensors are initially concentrated in the top-left corner of the field. The relocation paths show that the algorithm distributes sensors toward uncovered areas efficiently. The average movement distance is higher in the initial iterations due to the need to traverse a larger portion of the field, but it stabilizes quickly. The monitoring sensor count rises steadily, reaching near-optimal coverage despite the initial imbalance, showcasing the adaptability of the algorithm to corner-biased deployments.

Scenario 6: Top-Right Clustered Deployment

In Fig. 6, this scenario begins with sensors clustered in the top-right corner. The movement paths again show significant relocation as sensors disperse to cover the entire field. The average moving distance curve and the monitoring sensor curve follow trends similar to Fig. 5, with slightly different slopes reflecting the spatial characteristics of the initial deployment. The algorithm successfully achieves balanced coverage and convergence within a comparable number of iterations

Scenario 7: Multiple Cluster deployment

This is a more challenging scenario, sensors start in a tight cluster located centrally but not uniformly spread. The algorithm effectively disperses the sensors outward, as evidenced by the movement paths. The average moving distance is higher initially due to the compactness of the starting positions but declines rapidly as the sensors spread out. The number of monitoring sensors increases significantly within the first few iterations and then stabilizes, indicating efficient handling of clustered initial conditions.

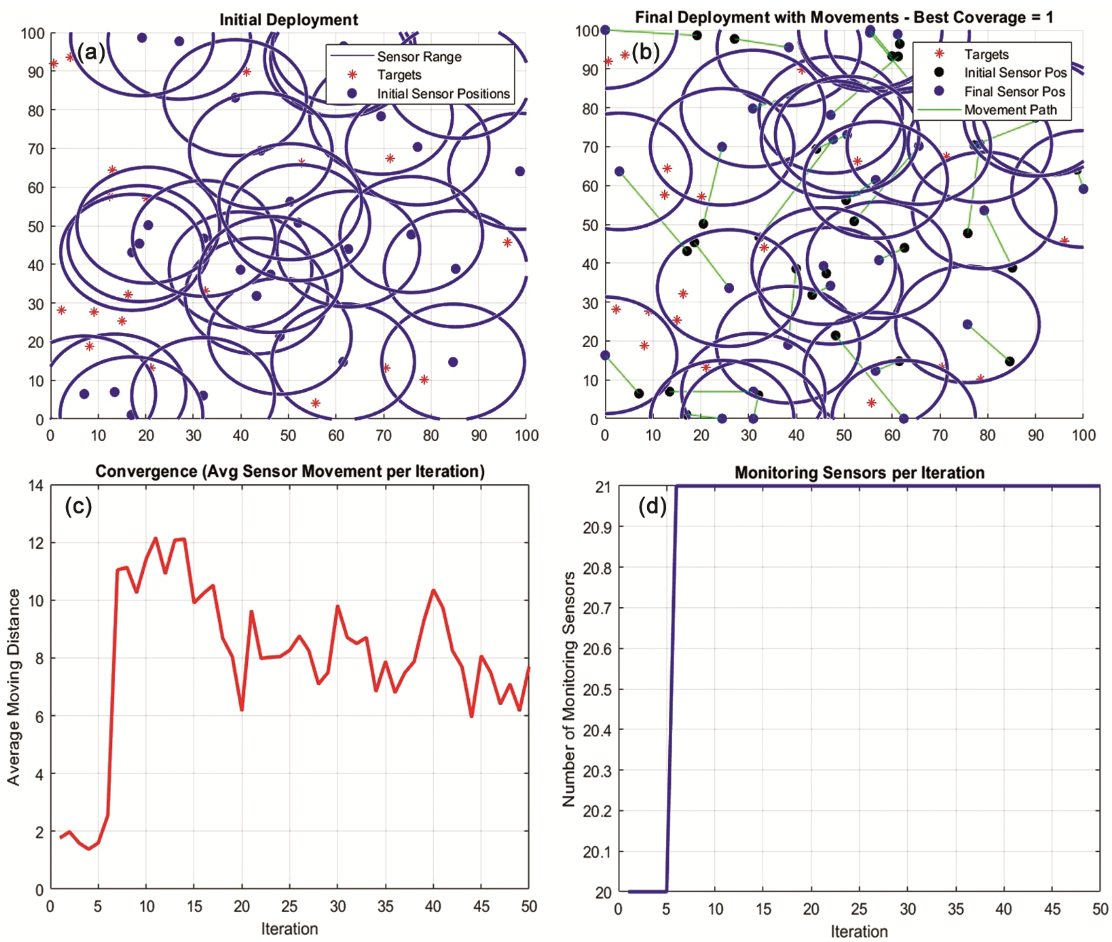

Fig. 1 — (a) Initial sensor positions (b) Final sensor positions with moving path (c) Average moving distance vs iteration number (d) Number of Monitoring sensor vs iteration number

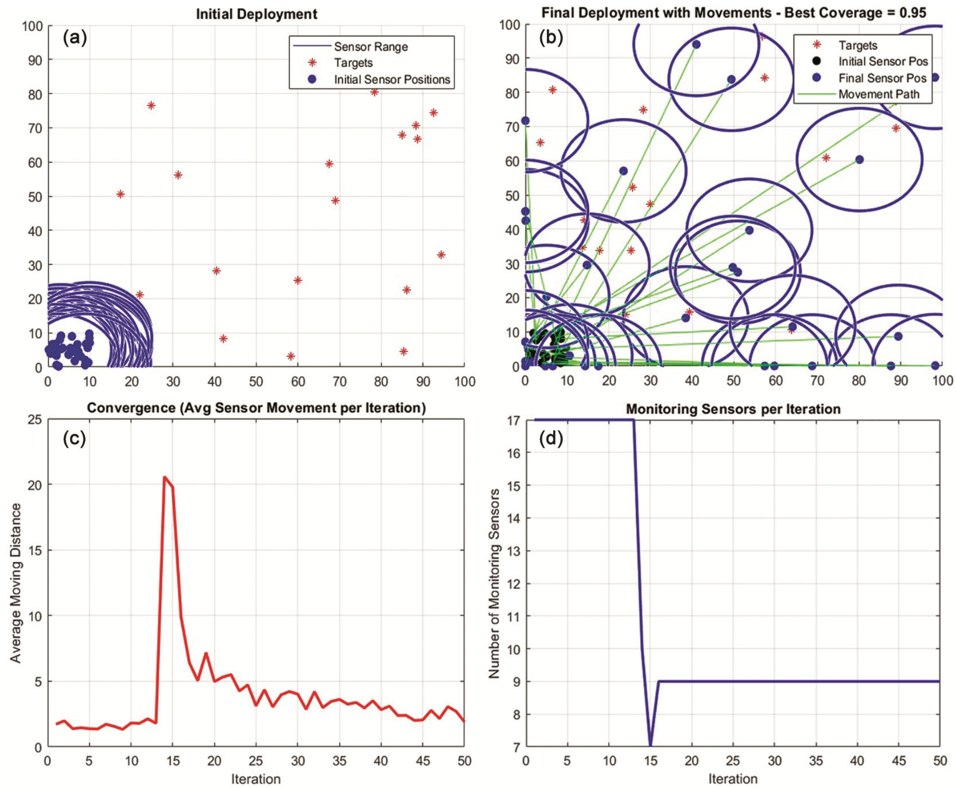

Fig. 2 — (a) Initial sensor positions (bottom-left deployment) (b) Final sensor positions with moving path (c) Average moving distance vs iteration number (d) Number of Monitoring sensor vs iteration number

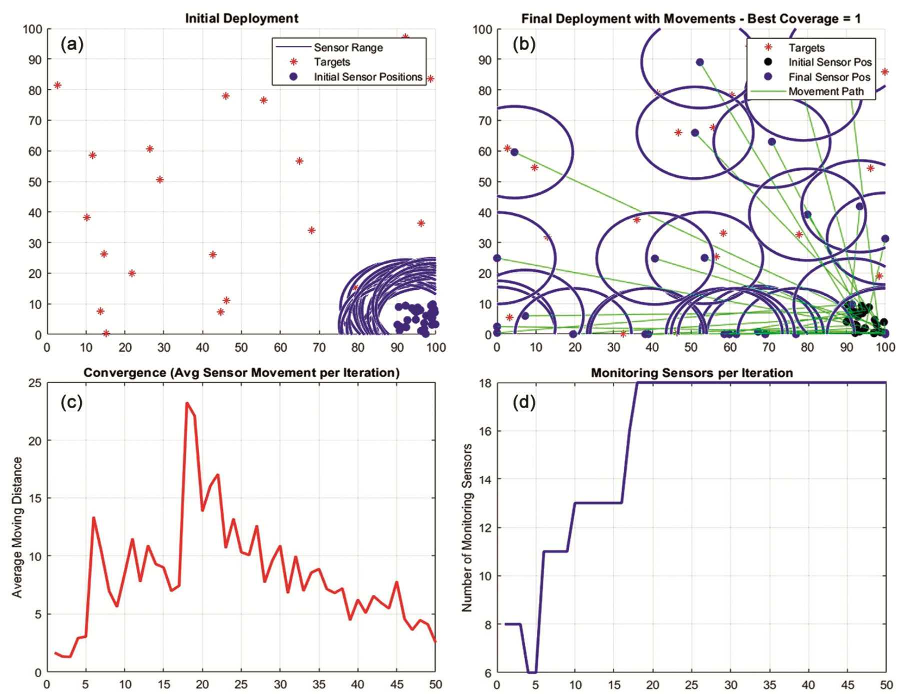

Fig. 3 — (a) Initial sensor positions (bottom-right deployment) (b) Final sensor positions with moving path (c) Average moving distance vs iteration number (d) Number of Monitoring sensor vs iteration number

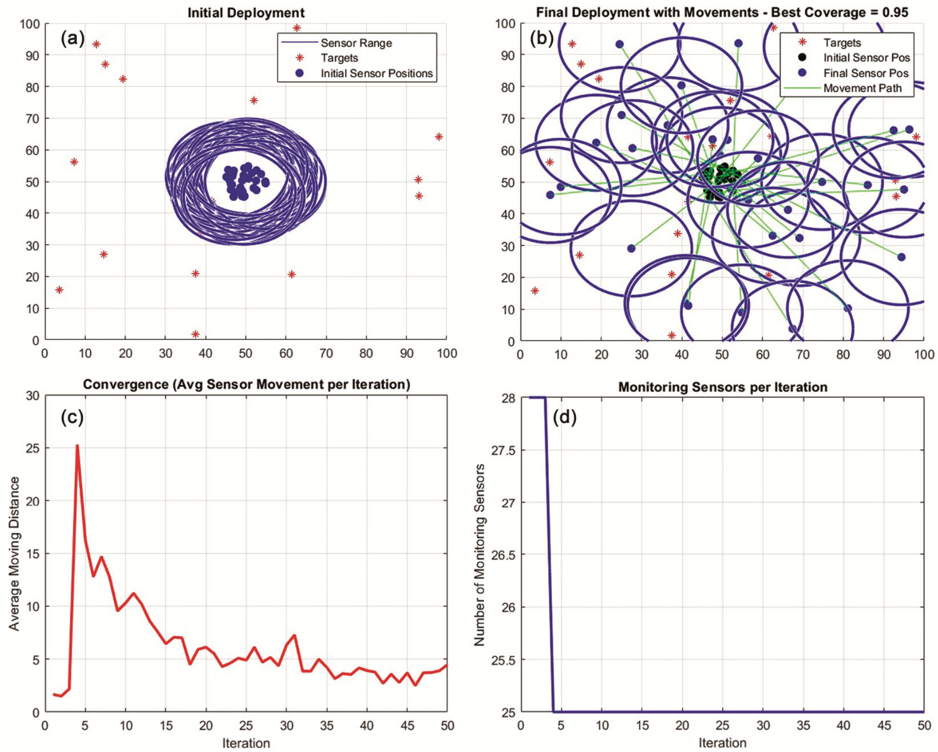

Fig. 4 — (a) Initial sensor positions (Center deployment) (b) Final sensor positions with moving path (c) Average moving distance vs iteration number (d) Number of Monitoring sensor vs iteration number

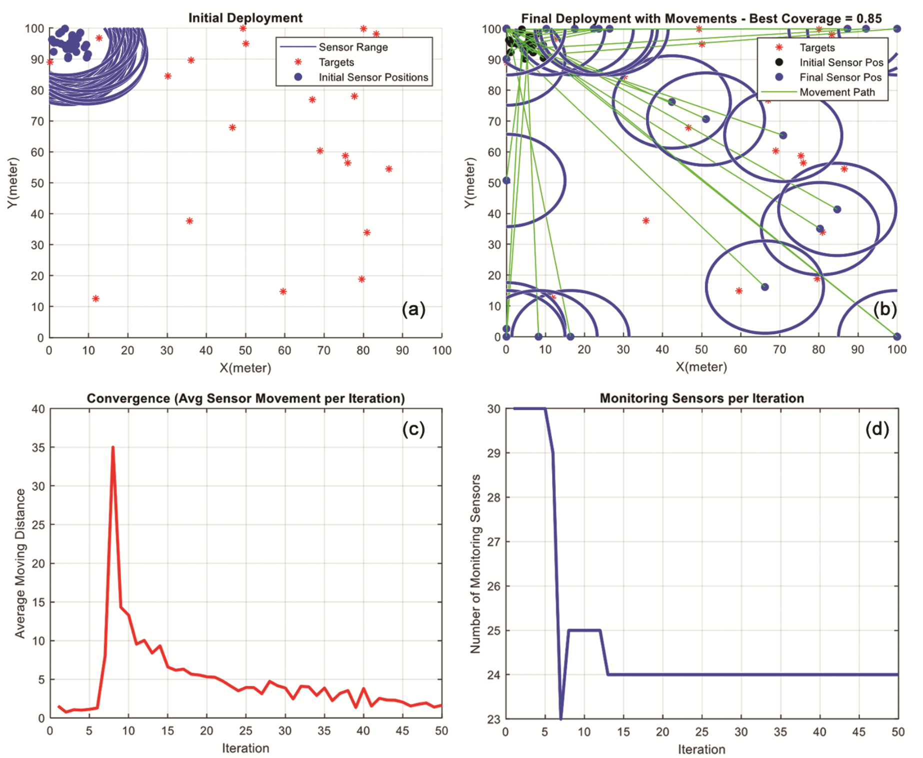

Fig. 5 — (a) Initial sensor positions (Top-left deployment) (b) Final sensor positions with moving path (c) Average moving distance vs iteration number (d) Number of Monitoring sensor vs iteration number

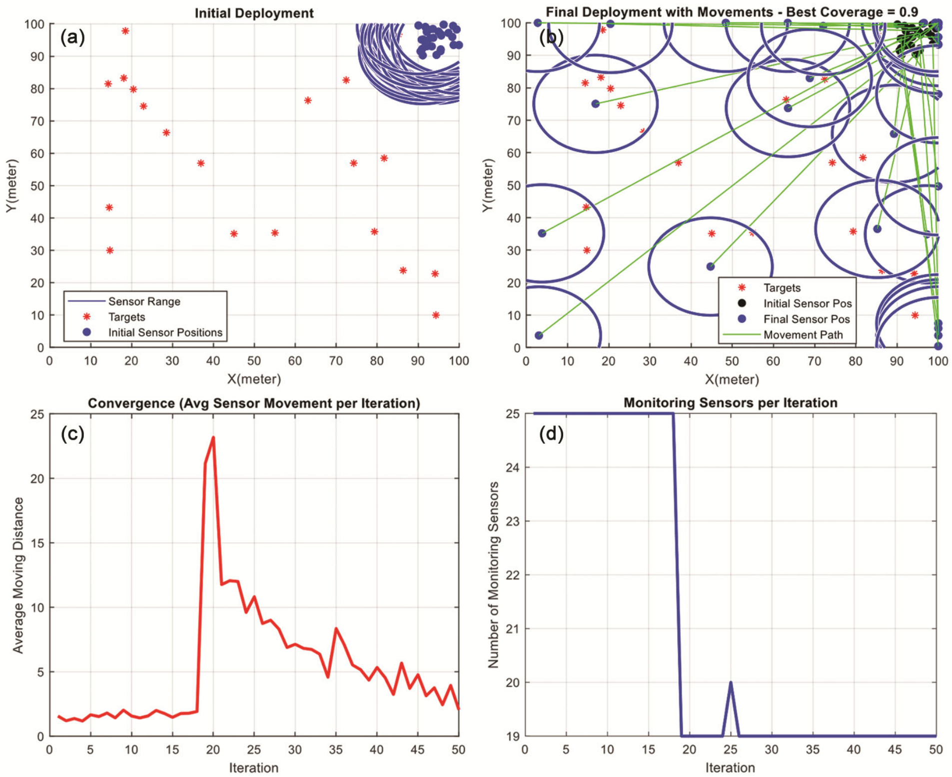

Fig. 6 — (a) Initial sensor positions (Top-right deployment) (b) Final sensor positions with moving path (c) Average moving distance vs iteration number (d) Number of Monitoring sensor vs iteration number

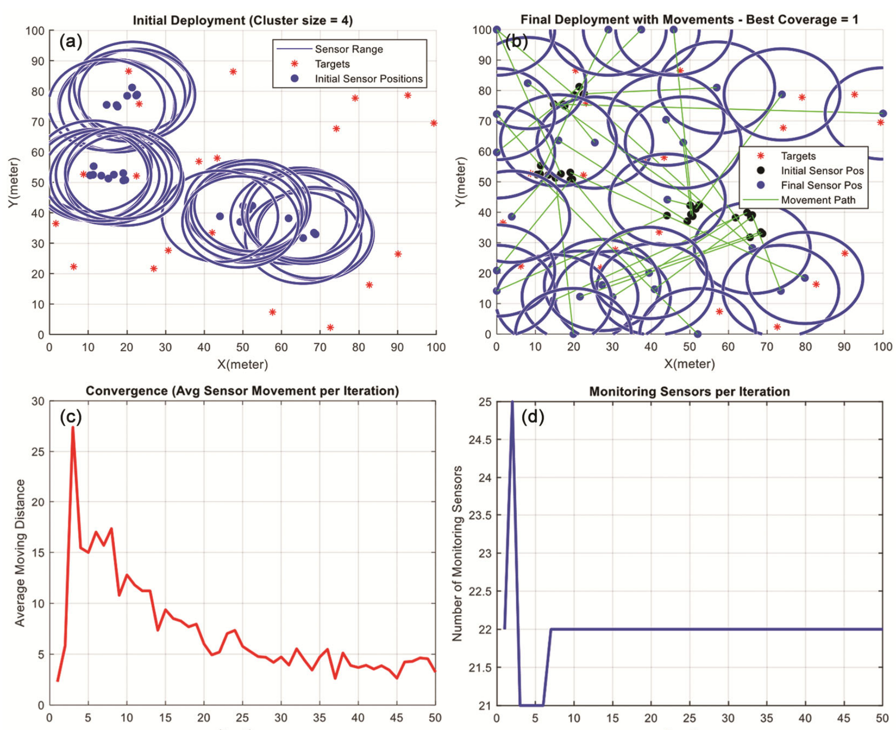

Fig. 7 — (a) Initial sensor positions (Multiple Cluster deployment) (b) Final sensor positions with moving path (c) Average moving distance vs iteration number (d) Number of Monitoring sensor vs iteration number

6 Conclusion

We investigate the target coverage of a randomly distributed mobile sensor network across different scenarios. This technique performs well in situations involving random deployment, as well as in bottom left, bottom right, top left, top right and centre deployments. The suggested approach maintains connection throughout the deployment process while minimizing the necessary number of sensor nodes, so assuring a substantial quantity of sensor nodes is accessible for target monitoring. The simulation results validate that the suggested hybrid VFA+PSO algorithm is both robust and efficient. It reliably adjusts to various initial deployment conditions, guarantees appropriate target coverage, and reduces unnecessary sensor movement. These features provide it an appropriate choice for practical wireless sensor network applications demanding dynamic and adaptive deployment methods.

This study is predicated on the premise of deterministic models in which sensing and communication ranges are represented as circular disks. However, in practical applications, this model seems unreal, and current research contends that probabilistic communication and sensing models are more suitable for accurately expressing reality. Consequently, we will examine probabilistic communication and sensing models as a component of our prospective research activities.

References

1 Ramin Y & Hashemi S N, *J Ambient Intelligence Humanized Computing,* 14 (5) (2023) 5941.

2 Mini S, Udgata S K & Sabat S L, *IEEE sensors J,* 14 (3) (2013) 636.

3 Cardei, Mihaela, *et al*, *Proceedings IEEE 24th Annual Joint Confer IEEE Comp Comm Societies,* 3 (2005).

4 Zou, Yi & Chakrabarty K, *ACM Trans Embedded Comp Syst,* 3 (1) (2004) 61.

5 Akram, Junaid, *et al*, *IEEE 92nd Vehicular Technol Confer IEEE*, 2020.

6 Katti & Anvesha, *J King Saud University-Comp Information Sci,* 34 (3) (2022) 734.

7 Liang, Dieyan, Shen H & Chen L, *Sensors,* 21 (1) (2020) 184.

8 Singh, Samayveer, *et al*, *IEEE Access,* 8 (2020) 74315.

9 Njoya, Arouna Ndam, *et al*, *Wireless Personal Comm,* 112 (2020) 1893.

10 Banoth, Rao S P, Donta P K & Amgoth T, *Wireless Networks,* 29 (4) (2023) 1815.

11 Arivudainambi D, Pavithra R & Kalyani P, *J Discrete Math Sci Cryptogr,* 24 (4) (2021) 975.

12 Manju, Bhambu P & Kumar S, *J Discrete Math Sci Cryptogr,* 23 (2) (2020) 651.

13 Mottaki A, Nemat, Motameni H & Mohamadi H, *Phys Comm,* 54 (2022) 101719.

14 Cheng, Chien-Fu & Wang C, *IEEE Trans Mobile Comp,* 17 (5) (2017) 1216.

15 Van den Bergh, Frans, Engelbrecht A P, *IEEE trans evol comput,* 8 (3) (2004) 225.

16 Binh, Huynh Thi Thanh, *et al*, *Evol Intelligence,* 17 (5) (2024) 3151.

17 Haris M & Nam H. *Sensors,* 25 (12) (2025) 3620.