Iraqi Journal of Information and Communications Technology(**IJICT**)
Vol. 8, Issue 2, August 2025
www.ijict.edu.iq
ISSN:2222-758X
e-ISSN:2789-7362

EFFICIENT COVERAGE OF SENSORS IN A WSN USING A MODIFIED HYBRID PSO AND ALO ALGORITHMS

Jafar S. Abdullah [1], **Mohammed I. Aal-Nouman** [2], **Belal Al-Fuhaidi** [3]
[1,2] Department of Information and Communication Engineering, College of Information Engineering, Al-Nahrain University, Jadriya, Baghdad, Iraq
[3] Faculty of Computing and IT (Computer Science), University of Science and Technology (UST), Yemen
jafar.saeed@coie-nahrain.edu.iq [1], m.aalnouman@nahrainuniv.edu.iq [2], belalarh@ust.edu.ye [3]
Corresponding Author: **Mohammed I. Aal-Nouman**
Received:26/11/2024; Revised:27/01/2025; Accepted:21/05/2025
DOI:**10.31987/ijict.8.2.309**

***Abstract*- The issue of sensor coverage in Wireless Sensor Networks (WSNs) is crucial, particularly as these networks are deployed in military applications for the armed forces as well as in civilian health applications. Therefore, improving coverage and communication while minimizing interference between sensors is essential. This paper presents a hybrid meta-heuristic approach to optimizing node deployment in WSNs using a modified Particle Swarm Optimization (mPSO) and Ant Lion Optimization (ALO) algorithms. The Particle Swarm Optimization (PSO) algorithm was applied for global search, while the ALO focused on internal search within the Region of Interest (ROI). Initially, nodes are deployed randomly within the ROI. The algorithm then detects uncovered gaps and iteratively enhances node placement, leading to an improved coverage ratio and minimized node overlap. The results of the hybrid meta-heuristic algorithm show improved performance compared to using PSO and ALO separately. This approach leads to an enhanced network lifetime and energy consumption of the WSN.**

keywords: **Wireless sensor networks (WSNs), Node deployment, Particle Swarm Optimization (PSO), Ant Lion Optimization (ALO), Coverage area, Overlapping nodes.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are increasingly recognized as essential tools for monitoring, inspection, and management across various applications. These include medical, agricultural, industrial, and also military applications. WSNs are typically composed of small sensors with limited Central Processing Unit (CPU) power, minimal storage capacity, restricted sensing and communication ranges, and low battery life, as shown in Fig. 1. Due to these constraints, WSNs often face challenges related to power management, coverage, network lifetime, and sensor-to-sensor communication. To cover the Region of Interest (ROI) with wireless sensor networks, each target area must be monitored by at least one sensor, with communication ensured between each sensor and the rest of the network, all the way to the sink node. Both coverage and communication are critical to network performance, as they enhance reliability and extend the network's lifetime [1-4].
There are two primary approaches to deploying sensors in the ROI. The first is the deterministic method, which involves analyzing the area beforehand to identify optimal sensor placement. While this approach typically yields superior coverage and performance, it requires significant effort and higher costs before deployment. The second approach is the random method, where sensors are deployed without prior analysis of the area. Although this method reduces deployment costs, it often leads to challenges such as coverage gaps, lack of connectivity between sensors, unbalanced energy consumption across the network nodes, and sensors overlapping. Therefore, optimizing sensor positions after deployment becomes essential to enhance network performance and efficiency [1, 2, 5].

ISSN:2222-758X
e-ISSN:2789-7362

Metaheuristic optimization algorithms are widely used to solve engineering problems and computational tasks due to their simplicity and practicality. Particle Swarm Optimization (PSO) and Ant Lion Optimization (ALO) are part of the family of swarm-based intelligent algorithms, inspired by the behaviors of animals as they search for food and interact with one another to achieve their goals Metaheuristic optimization is used to find optimal or near-optimal solutions to complex tasks, particularly when conventional methods are inefficient or too slow [6, 7].

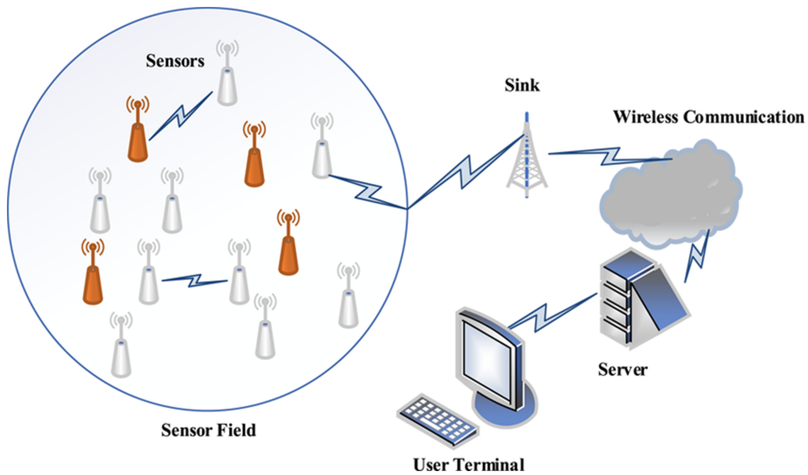


Figure 1: The structure of WSN [8].

This paper addresses the problems of random sensor deployment by hybrid meta-heuristic optimization algorithms. The main key contributions are:

- identifying uncovered gaps in the ROI and covering the area using a mobile node.
- enhancing network energy consumption and extending network lifetime by selecting the nearest mobile sensor to cover the holes.
- reducing overlapping nodes and areas by repositioning the node to the optimal location within the ROI.
- increasing the overall coverage ratio of the network.

The remainder of this paper is organized as follows: Section II reviews the most relevant related works. In Section III, presents and discusses the problem statement. The proposed optimization algorithm is summarized in Section IV. Section V presents the experimental simulation results. Finally, Section VI provides the conclusion and suggestions for future work.

II. RELATED WORKS

In recent years, many optimization algorithms have been applied to address the coverage problem in WSNs, and multiple algorithms have been combined to achieve better results.

In [8], a novel hybrid metaheuristic optimization algorithm called MOFAC-GA-PSO was proposed, combining the Genetic Algorithm (GA) and PSO to minimize overlap and achieve full area coverage in the ROI using guided random deployment methods. This approach required multiple pre-initial deployment steps, leading to additional costs for studying the target

ISSN:2222-758X
e-ISSN:2789-7362

area before deployment, and it may not be suitable for all environments. The authors used GA for global search and PSO for local search.

In [9], the Integrated Learning Particle Cluster Optimization (ICLPSO) algorithm was introduced to achieve a high coverage rate in the region of interest, reaching 97% in the experiments. However, the overlap between sensor nodes was relatively high, leading to the use of more sensors, irregular energy consumption, and redundant information at the sink. In [9] the ALO algorithm was combined with the Blocked Search (TS) algorithm, referred to as ALO-TS. The authors used ALO as the global optimizer and TS as the local optimizer, achieving good results in increasing coverage in the region of interest. However, node interference was not considered, resulting in significant overlap between sensors and some gaps between them. In [10] an improved adaptive Salp Swarm Algorithm (SSA) was proposed, incorporating T-distribution mutation with a chaotic sequence algorithm to enhance the algorithm's local search ability. However, the results were poor due to a low coverage rate and significant overlap, especially with a high number of search iterations in the algorithm.

In [11] an Improved Weed Optimization (IWO) algorithm was proposed to solve the coverage problem using a hybrid strategy. This approach yielded good results, improving both coverage and convergence speed while addressing the issue of falling into local optima compared to other algorithms.

III. WIRELESS SENSOR NETWORK MODEL

Suppose a sensor s_i is deployed within a two-dimensional ROI at coordinates (x_i, y_i). A target grid point p is located at coordinates (x, y). To calculate the coverage ratio, if the distance between sensor s_i and point p is within the sensor's range, point p is considered covered. The distance can be calculated using the Euclidean distance formula, represented as following Eq. (1):

$$d(s_i, p) = \sqrt{(s_i(x_i) - p(x))^2 + (s_i(y_i) - p(y))^2} \tag{1}$$

The sensibility of any sensor s_i in the ROI to point p can be defined by the Eq. (2), indicating a reverse relationship between the sensibility of sensor and the Euclidean distance.

$$S(s_i, p) = \frac{\delta}{(d(s_i, p))^k} \tag{2}$$

$S(s_i, p)$ refers to the sensibility of node, $d(s_i, p)$ refer to the Euclidean distance and δ and k represent fixed value specific to the sensor [12, 13].

A. *Coverage Models*

The coverage models of WSNs are divided into two main types: the Boolean coverage model and the Probabilistic coverage model. In this paper, the Boolean coverage model is used for each sensor node within the ROI [14].

Assume all nodes have a sensing range r_s and a communication range R_c. The sensing range is represented as a circular disk around the sensor and is used to detect and record events, while the communication range ensures connectivity between sensors within the WSN. The communication radius of a node is generally greater than or equal to twice its sensing radius, as shown in Fig.2 [15].

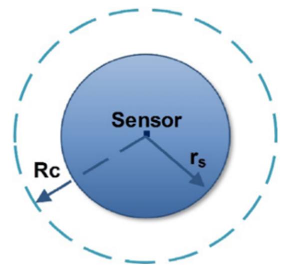


Figure 2: Sensing range r_s and communication range R_c [2].

$$C_{x_iy_i}(s_i) = \begin{cases} 1, & \text{if } d(s_i, p) < r_s, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The coverage of a sensor node $C_{x_iy_i}$, based on the Boolean coverage model, is represented by a positive value when the Euclidean distance $d(s_i, p)$ between the point and the sensor is greater than or equal to sensing radius r_s as defined by Eq. (3) [13, 16].

To compute the overall coverage of a WSN, as shown in Eq. (4), the sum of the probabilities of covered grid points within the ROI are calculated and divided by the total number of grid points.

$$C_{\text{wsn}} = \frac{\sum C_{x_iy_i}(s_i)}{N} \tag{4}$$

As the number of grid points in the area increases, the computation task of coverage becomes more complex and requires more processing within the optimization algorithm.

B. Suppositions

In this paper, certain assumptions have been taken into account for optimizing coverage. These include:

- The ROI for deployment is considered a two-dimensional space.
- The number of sensor nodes is fixed and determined before starting the experiments.
- All sensor nodes have the same sensing range r_s and a communication range r_c within the ROI, represented as a circular disk around each node.
- Sensors can move freely within the ROI without any obstacles.
- All nodes are aware of the locations of other sensors within the ROI.

IV. OPTIMIZATION ALGORITHM

Nature-inspired optimization algorithms are highly effective tools for solving computational optimization problems by finding the best or most optimal solutions within a given workspace. Each algorithm should have exploration and exploitation processes to find both the global and local optimum solutions. The exploration process searches across far positions to find the global best solution, while the exploitation process focuses on neighboring locations to identify the local best solution [17, 18].

ISSN:2222-758X
e-ISSN:2789-7362

A. Overview Particle Swarm Optimization algorithm

The PSO algorithm is considered a simple yet powerful tool within the field of meta-heuristic algorithms, imitating the behavior of bird swarms and fish schools moving together to find food or to avoid the enemy. In PSO, each particle represents a sensor location within the ROI and is initially deployed randomly, in algorithm setting the total number of particles to N. These particles continue adjust position in workspace to find an optimal best position based on the fitness function criteria. To implement PSO, Eqs. (5 and 6) are used. Let x_i represent the position of a particle, defined as $x_i = [x_1, x_2, \ldots, x_d]$, and v_i denotes the velocity of each particle in the ROI, represented as $v_i = [v_1, v_2, \ldots, v_d]$. Additionally, p_i refers to the local best position, and p_g refers to global best position for all particles [13, 19].

$$v_i(t+1) = \omega \times v_i(t) + c_1 \times r_1 \left(p_i - x_i(t)\right) + c_2 \times r_2 \left(p_g - x_i(t)\right) \tag{5}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{6}$$

In Eq. (5), ω refers to inertia constant value, c_1 and c_2 are represent the cognitive and social constants, respectively, and r_1 and r_2 are random variables ranging from 0,1 [20].

B. Overview Ant Lion Optimization algorithm

The ALO algorithm is one of the latest nature-inspired algorithms, introduced by Mirjalili [21] in 2015. It is based on the behavior of antlion in nature. Antlions are considered predatory and dangerous insects, primarily feeding on other insects. They build cone-shaped traps by moving in a circular path and locating themselves at the center of the pits, waiting for prey to come and catch them. If the prey tries to escape, the antlion throws sand at it to make it fall to the bottom of the pit, making it easier to catch. After finishing with the prey, the ant discards its remains outside the trap, waiting for the next prey [21-24].

The mathematical equations of the ALO algorithm can be summarized in four stages: Step 1 describes the movement of the ants; Step 2 covers the ants falling into the traps; Step 3 shows the ants sliding toward the antlions; and finally, Step 4 involves the ants being caught, as following steps:

1) Ants Movemet:

   To represent the random movement of ants toward the antlion pits within the workspace used in the ALO algorithm according to the following Eq. (7):

$$X(t) = \left[0,\ \mathrm{cumsum}\big(2r(t_1) - 1\big),\ \mathrm{cumsum}\big(2r(t_2) - 1\big),\ \ldots,\ \mathrm{cumsum}\big(2r(t_{\max}) - 1\big)\right] \tag{7}$$

   cumsum refers to the cumulative sum, t represents the step size of random movement, and t_max is the maximum number of iterations in the algorithm. The value of $r(t)$ can be 1 or 0 based on a random value generated within the range [0,1]: if the random value is greater than 0.5, $r(t)$ equals 1; if it is less than or equal to 0.5, $r(t)$ equals 0. Additionally, iteration and workspace boundary limits are used to prevent random movement outside the ROI.

2) Ants Falling Into The Traps:

   Ants continue moving randomly until they fall into the antlion's pit, and at each iteration, the boundaries are updated

ISSN:2222-758X
e-ISSN:2789-7362

according to the following Eqs. (8 and 9):

$$c_i^t = \text{antlion}_j^t + c^t \tag{8}$$

$$d_i^t = \text{antlion}_j^t + d^t \tag{9}$$

c_i^t refers to minimum variables at t-th ants, d_i^t denotes the maximum variables at t-th ants. antlion_j^t represents the position of the selected or nearest antlion traps, while the d^t and c^t refer to maximum variables and minimum variables at i-th iteration.

3) Ants Sliding Toward The Antlions:
When an ant falls into the pit, the antlion throws sand to drive it to the bottom of the trap, making it easier to catch. This process limits the ant's movement boundary, gradually decreasing it. The following Eqs. (10 and 11) that describe this sliding motion:

$$c^t = \frac{c^t}{I} \tag{10}$$

$$d^t = \frac{d^t}{I} \tag{11}$$

Where, c^t represent the minimum variables at t-th iteration, d^t denotes the maximum variables at t-th iteration, and I represent the rate value, defined as:

$$I = 10^{\frac{w \cdot t}{T}} \tag{12}$$

Where, T represents the maximum number of iterations, t is the current iteration, and w is a constant used to control the exploitation process in ALO. The value of w changes as follows: $w = 2$ when $t > 0.1T$, $w = 3$ when $t > 0.5T$, $w = 4$ when $t > 0.75T$, $w = 5$ when $t > 0.9T$, and $w = 6$ when $t > 0.95T$.

4) Ants Being Caught
In the final stage, the ant is caught by the antlion, and based on a comparison of their criteria of fitness function, their positions are switched according to the following Eq. (13):

$$\text{antlion}_i^t = \begin{cases} \text{ant}_i^t, & \text{if } F(\text{ant}_i^t) > F(\text{antlion}_i^t) \end{cases} \tag{13}$$

After that, the waste is disposed of, and the trap is reset for the next ant.

V. PROPOSED MODIFIED PSO-ALO

The modified hybrid Particle Swarm Optimization and Ant Lion Optimization (mPSO-ALO) algorithms efficiently address coverage issues in WSN. The migration between the two algorithms is consistent, as they belong to the same meta-heuristic family. In the proposed algorithm, PSO is used for global search, while ALO is used for local search to help prevent the algorithm from converging to a local optimum solution, as in Fig. 3. Initially, sensors are randomly deployed in the ROI, leading to overlapping sensors and uncovered gaps. The mPSO algorithm is used to detect these gaps by communicating sensors and sharing their locations, allowing for the accurate identification of uncovered areas between nodes. The algorithm then searches for the nearest overlapping sensor to reposition it and cover the gap, reducing the randomness of sensor

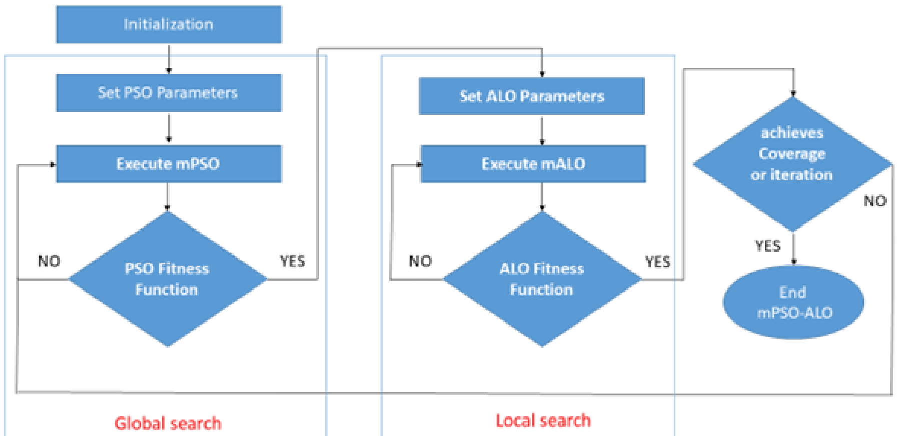


Figure 3: mPSO-ALO flowchart.

movement in the ROI. Additionally, ALO is used to locally search for the optimal sensor placement, ensuring effective gap coverage and achieving the highest possible coverage rate in the area. The main steps of the process are defined as follows:

1) Initialization: all variables required for mPSO-ALO are initialized and the sensors are randomly deployed in the ROI.
2) Set PSO Parameters: the local and global best positions, and initialize particle positions and velocities are defined.
3) Execute mPSO: gaps are calculated and particle positions and velocities are updated to find the optimal location.
4) PSO Fitness Function: If the fitness of the new position is better than the previous one, ALO is proceeded to; otherwise, mPSO execution is repeated.
5) Set ALO Parameters: the positions of ants and antlions traps are defined.
6) Execute mALO: ant movement and antlion traps are calculated to improve the best sensor positions.
7) ALO Fitness Function: If the fitness of the new position is better than the previous one, the next steps is proceeded to; otherwise, mALO execution is repeated.
8) Coverage Check: If the network has reached the required coverage, stop; otherwise, repeat from step 3.

A. Objective function

To enhance sensor node positions and increase the coverage ratio within the ROI, several criteria are used in the fitness function of the modified optimization algorithm. The following are the main criteria used in the objective function in mPSO-ALO:

- Coverage Rate: Sensors should be placed in optimal positions to maximize area coverage and increase the network's coverage rate.
- Uncovered Holes: Detects gaps in the area that need coverage, aiming to cover each gap with at least one sensor to improve the WSN and enhance connectivity between sensors.

- Overlapping Nodes: Full overlap is generally undesirable in the network, but partial overlap among sensors is acceptable. Additionally, overlap should be minimized in the ROI.
- Energy Consumption: Selecting the nearest sensor to cover gaps or reduce full overlap helps minimize power consumption and extend network lifetime.

The more complex the fitness function, the longer the algorithm's execution time. In the proposed algorithm, this function is used in both PSO and ALO. Therefore, a balance between complexity, speed, and performance must be maintained.

VI. SIMULATION RESULTS

A. Simulation Settings

The mPSO-ALO algorithm was implemented and analyzed using Matrix Laboratory (MATLAB) program v2023, running on a laptop with a Core i7 processor and 8 GB of Random Access Memory (RAM). The inertia parameter, along with cognitive and social constants for the modified Particle Swarm Optimization algorithm, were applied and tested multiple times across different scenarios to determine the optimal values for achieving good results in the coverage optimization problem.

To test the proposed mPSO-ALO algorithm of this work, an experiments was conducted using different scenarios with varying numbers of sensor nodes, recording the results and comparing them. Table I lists the main parameter settings required for the experiments.

TABLE I
EXPERIMENTAL PARAMETER SETTINGS

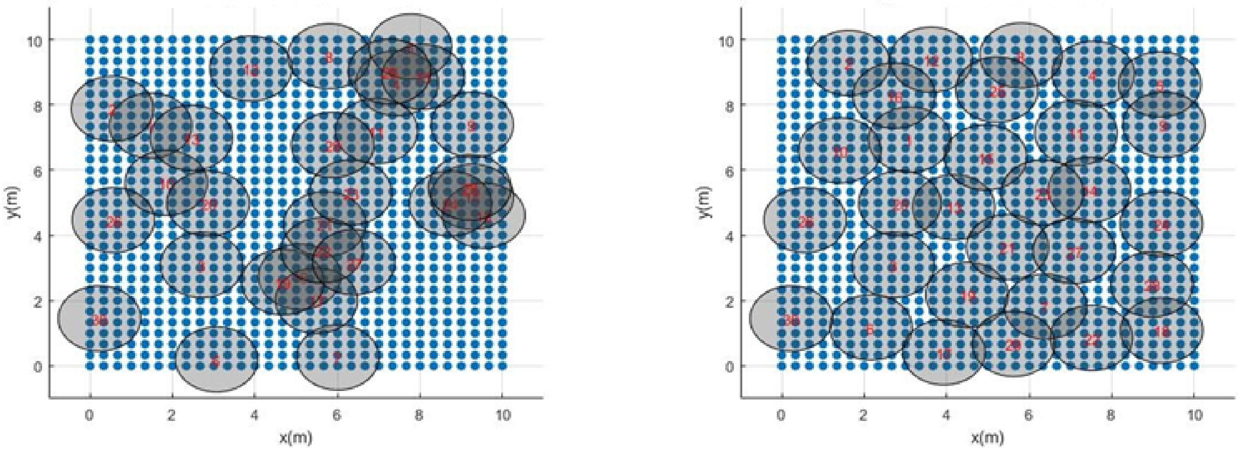
No. of Sensors	Area	Sensing Range	No. of Iterations	Communication Range	mPSO Parameter	No. of Points
30					$\omega = 0.8$	
35	10×10 m^2	1 m	300	2 m	$c_1 = 0.9$	1000
40					$c_2 = 0.9$	

B. Simulation Results

Initially, 30 sensor nodes were randomly deployed within a 10×10 m^2 area, covering 57% of the ROI as in Fig. 4a. Subsequently, the PSO algorithm was applied, increasing the coverage rate to 78% as in Fig. 4b. Following this, the ALO algorithm was used, further enhancing the coverage rate to 83%, as in Fig. 5a. Finally, the proposed mPSO-ALO algorithm was implemented, achieving a coverage 92%, as in Fig. 5b.

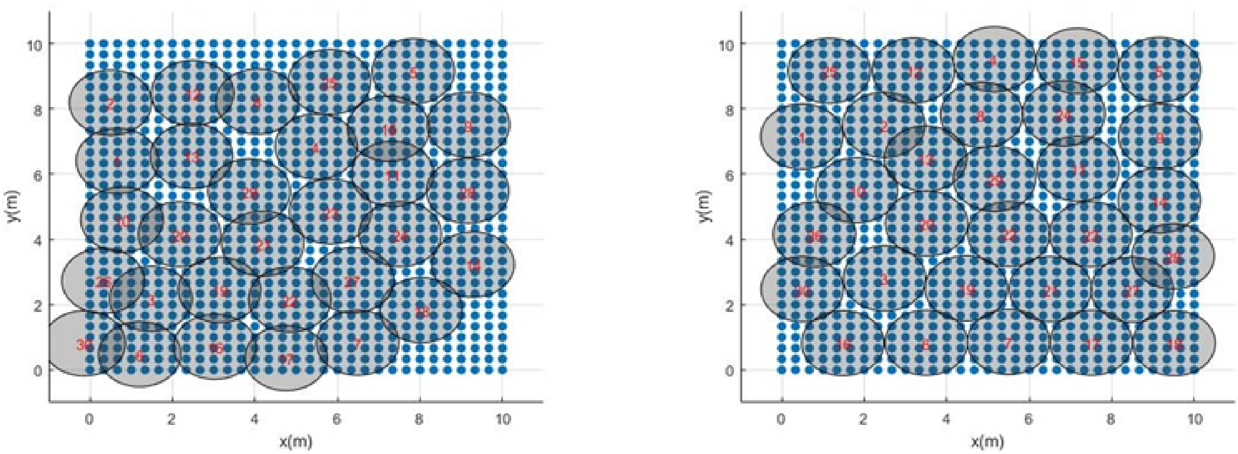
In the second test, an initial random deployment of 35 sensor nodes in a 10x10 m^2 area resulted in a 66% coverage of the ROI, as in Fig.6.a. To optimize coverage, then applied the PSO algorithm, as in Fig. 6b, the ALO algorithm, as in Fig. 7a, and our proposed mPSO-ALO algorithm, as in Fig. 7b. PSO increased coverage to 86%, ALO to 88%, and mPSO-ALO achieved a significant improvement, reaching 98% coverage.

www.ijict.edu.iq
ISSN:2222-758X
e-ISSN:2789-7362



(a) random deployment (b) PSO

Figure 4: Deployment 30 sensors within ROI in random method and applied PSO algorithm.

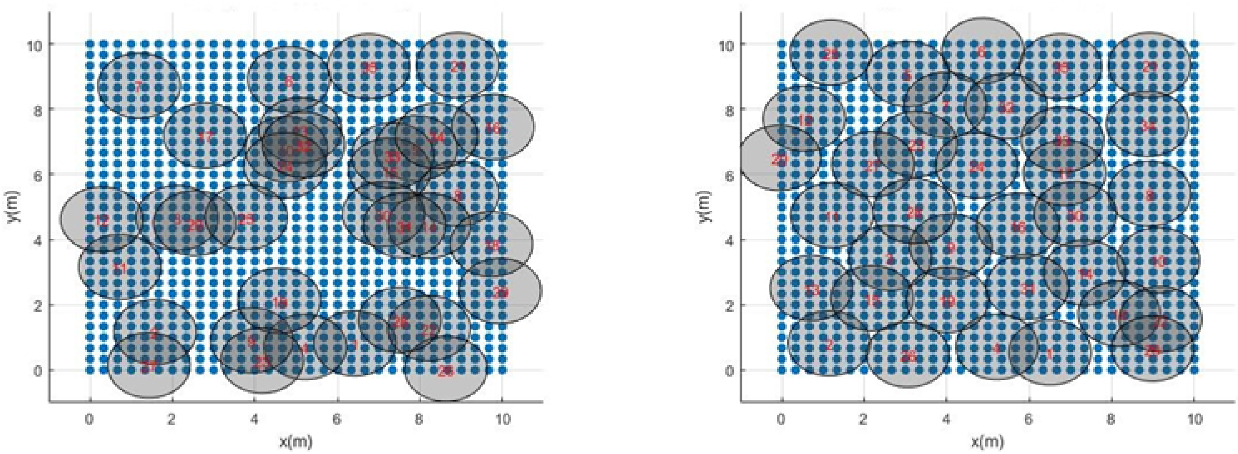


(a) ALO (b) mPSO-ALO

Figure 5: Deployment 30 sensors within ROI by applied ALO and by applied mPSO-ALO algorithms.

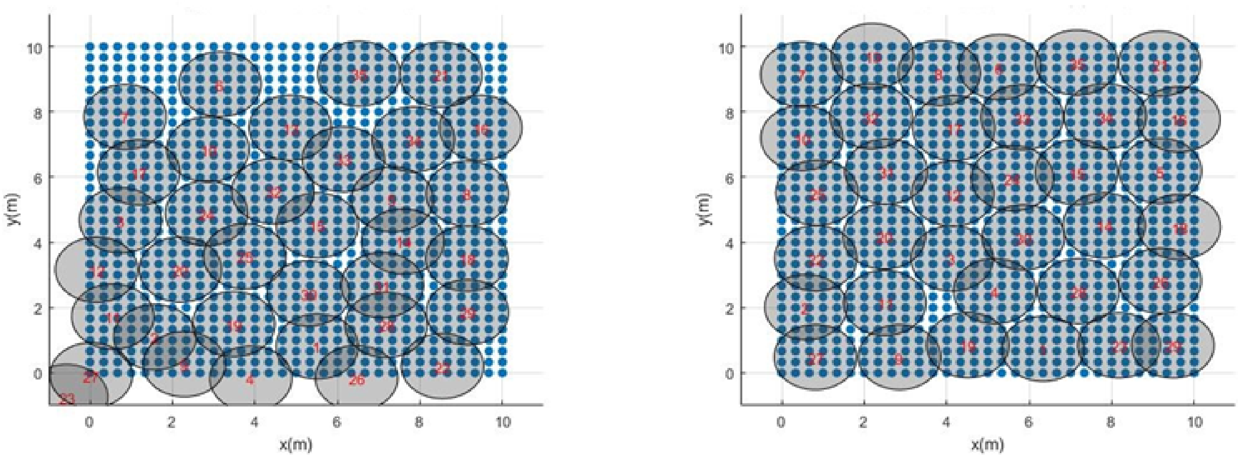
In the third experiment, 40 sensor nodes were randomly deployed within the ROI, initially covering 67%, as shown in Fig. 8a. The PSO algorithm was then applied, raising the coverage ratio to 87%, as illustrated in Fig. 8b. Next, the ALO algorithm further improved the coverage ratio to 92%, as shown in Fig. 9a. Finally, the modified mPSO-ALO algorithm (proposed in this paper) was applied, achieving a full coverage rate of 100%, as depicted in Fig. 9b.

ISSN:2222-758X
e-ISSN:2789-7362
www.ijict.edu.iq



(a) random deployment (b) PSO

Figure 6: Deployment 35 sensors within ROI in random method and applied PSO algorithm.



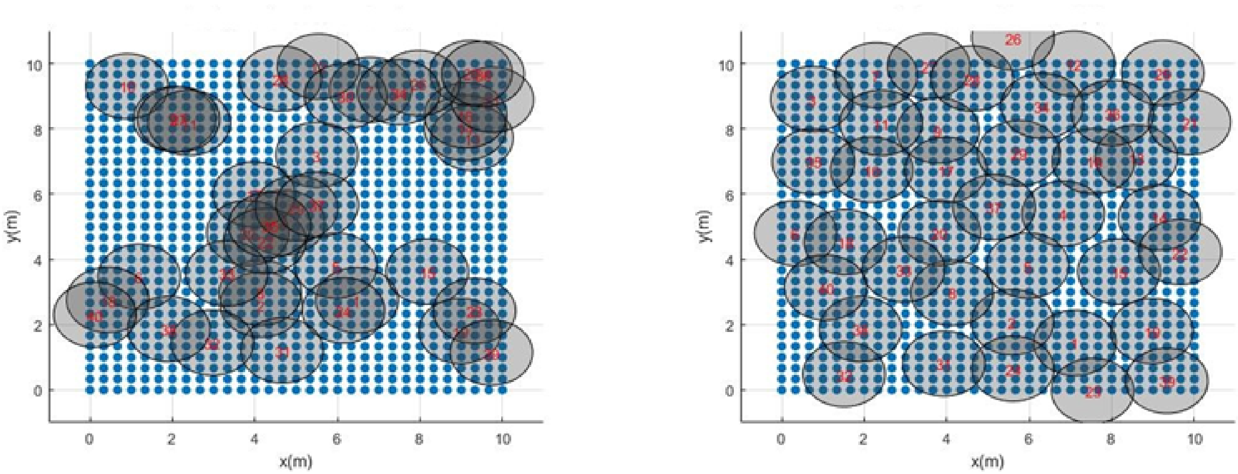
(a) ALO (b) mPSO-ALO

Figure 7: Deployment 35 sensors within ROI by applied ALO and by applied mPSO-ALO algorithms.

C. Discussion results

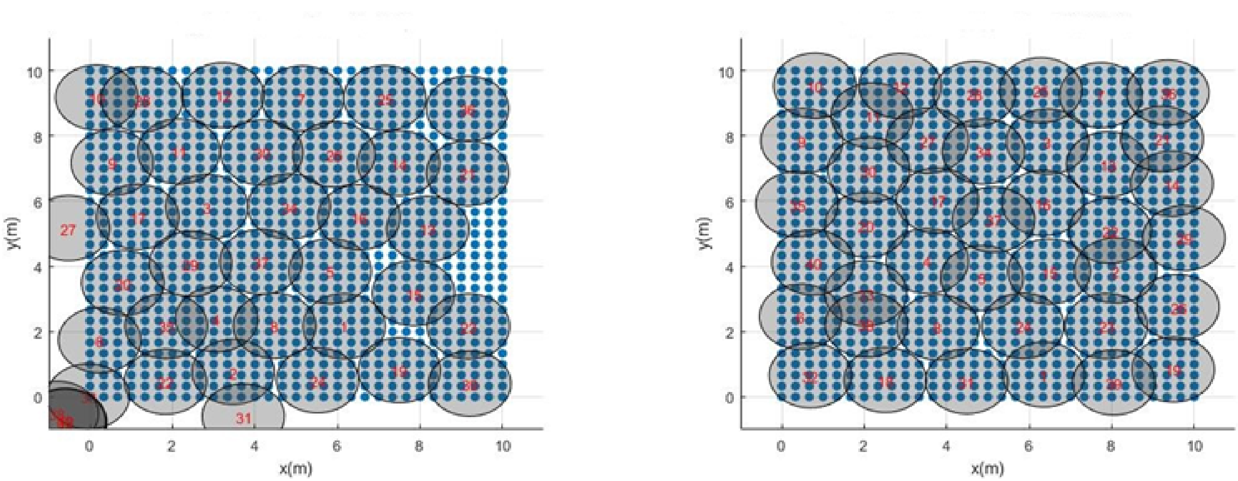
The results of the three experiments above indicate that while the PSO and ALO algorithms achieve relatively high coverage as illustrated in Table III and Fig. 10, but they do not produce the optimal sensor placement within the ROI. This is because they rely only on random sensor movement without considering other critical factors. For instance, Sensors should avoid moving far from their initial deployment areas unless necessary to conserve node energy. They should also be capable of detecting gaps within the ROI and selecting the nearest gaps to move toward. The ALO algorithm tends to move sensors in smaller areas, whereas the PSO algorithm covers larger areas. These aspects were addressed in our

www.ijict.edu.iq
ISSN:2222-758X
e-ISSN:2789-7362



(a) random (b) PSO

Figure 8: Deployment 40 sensors within ROI in random method and applied PSO algorithm.



(a) ALO (b) mPSO-ALO

Figure 9: Deployment 40 sensors within ROI by applied ALO and by applied mPSO-ALO algorithms.

mPSO-ALO algorithm, where enhanced the fitness function to detect and handle gap locations, sensor overlapping, select the nearest node to covered holes, and balanced movement between global and local search, ultimately achieving optimal coverage.

ISSN:2222-758X
e-ISSN:2789-7362

TABLE II
COVERAGE RATE SUMMARY

No. of Sensors	Coverage Randomly	Coverage PSO	Coverage ALO	Coverage mPSO-ALO	Increased Ratio
30	57%	78%	83%	92%	61%
35	66%	86%	88%	98%	48%
40	67%	87%	92%	100%	49%

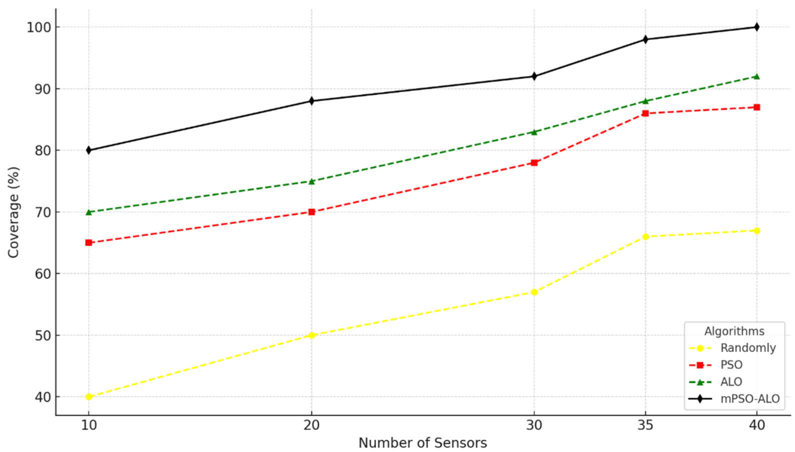


Figure 10: Compare coverage rate with numbers of sensor.

VII. CONCLUSIONS AND FUTURE WORK

In this study, the PSO algorithm, ALO algorithm, and the proposed mPSO-ALO algorithm were applied across three scenarios in MATLAB program to optimize sensor node placement within the ROI. The coverage rates achieved by each algorithm were compared to those of the PSO and ALO algorithms. The results demonstrate that the proposed algorithm achieves the highest coverage rate with minimal overlap, effectively covering all uncovered gaps in the ROI to ensure optimal connectivity between sensors. Additionally, it reduces node energy consumption, enhances network lifetime, and accomplishes this with a minimal number of sensor nodes. Moreover, the proposed mPSO-ALO algorithm efficiently balances local and global search (exploration and exploitation) and overcomes the tendency to fall into local optimum solutions.

For future work, this paper suggest to investigate the following points:

1) Introducing obstacles within the ROI and optimizing sensor deployment to avoid obstacles, thereby simulating a real-world environment.
2) Expanding the deployment area from 2D to 3D to simulate real-world deployment challenges in multi-level buildings and floors.
3) Integrate localization techniques alongside the deployment algorithm to identify neighboring nodes and detect gaps using the localization algorithm.

FUNDING

www.ijict.edu.iq
ISSN:2222-758X
e-ISSN:2789-7362

None.

ACKNOWLEDGEMENT

The author would like to thank the reviewers for their valuable contribution in the publication of this paper.

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] R. Priyadarshi, B. Gupta, and A. Anurag, "Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues," Journal of Supercomputing, vol. 76, no. 9, pp. 7333–7373, Sep. 2020, doi: 10.1007/s11227-020-03166-5.
[2] M. Farsi, M. A. Elhosseini, M. Badawy, H. Arafat Ali, and H. Zain Eldin, "Deployment techniques in wireless sensor networks, coverage and connectivity: A survey," IEEE Access, vol. 7, pp. 28940–28954, 2019, doi: 10.1109/ACCESS.2019.2902072.
[3] R. A. Al-Qassab and M. I. Aal-Nouman, "Performance Evaluation of CoAP and MQTT-SN Protocols," in Emerging Technology Trends in Internet of Things and Computing, Eds., Cham: Springer International Publishing, 2022, pp. 223–236, doi:10.1007/978-3-030-97255-4_17.
[4] S. K. Mudhafar and A. E. Abdelkareem, "Underwater Localization and Node Mobility Estimation," International Journal of Electrical and Computer Engineering, vol. 12, no. 6, pp. 6196–6209, Dec. 2022, doi:10.11591/ijece.v12i6.pp6196-6209.
[5] O. A. Abdullah, M. Aal-nouman, and A. Al-joudi, "Indoor WLAN localization via adaptive Lasso Bayesian inference and convex optimization," Cogent Eng, vol. 7, no. 1, Jan. 2020, doi: 10.1080/23311916.2020.1813677.
[6] K. E. Adetunji, I. W. Hofsajer, A. M. Abu-Mahfouz, and L. Cheng, "A Review of Metaheuristic Techniques for Optimal Integration of Electrical Units in Distribution Networks," IEEE Access, vol. 9, pp. 5046–5068, 2021, doi:10.1109/ACCESS.2020.3048438.
[7] A. A. Hussein and R. A. Khalid, "An Improved Energy Efficient Chain-Based Routing Protocol Using ACO", Iraqi Journal of ICT, vol. 2, no. 3, pp. 11–29, Sep. 2019, doi: 10.31987/ijict.2.3.75.
[8] S. Kumar De, A. Banerjee, K. Majumder, K. Kotecha, and A. Abraham, "Coverage Area Maximization Using MOFAC-GA-PSO Hybrid Algorithm in Energy Efficient WSN Design," IEEE Access, vol. 11, pp. 99901–99917, 2023, doi: 10.1109/ACCESS.2023.3313000.
[9] H. Ling, T. Zhu, W. He, H. Luo, Q. Wang, and Y. Jiang, "Coverage Optimization of Sensors under Multiple Constraints Using the Improved PSO Algorithm," Math Probl Eng, vol. 2020, 2020, doi: 10.1155/2020/8820907.
[10] H. Deghbouch and F. Debbat, "Dynamic Deployment of Wireless Sensor Networks using Hybrid Ant Lion Optimizer and Tabu Search," in ISIA 2020 - Proceedings, 4th International Symposium on Informatics and its Applications, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/ISIA51297.2020.9416541.
[11] J. Wang, Z. Zhu, F. Zhang, and Y. Liu, "An Improved Salp Swarm Algorithm for Solving Node Coverage Optimization problem in WSN," Apr. 27, 2023. doi: 10.20944/preprints202304.1025.v1.
[12] F. Zhu and W. Wang, "A Coverage Optimization Method For WSNs Based On The Improved Weed Algorithm," Sensors, vol. 21, no. 17, Sep. 2021, doi: 10.3390/s21175869.
[13] H. ZainEldin, M. Badawy, M. Elhosseini, H. Arafat, and A. Abraham, "An improved dynamic deployment technique based-on genetic algorithm (IDDT-GA) for maximizing coverage in wireless sensor networks," J Ambient Intell Humaniz Comput, vol. 11, no. 10, pp. 4177–4194, Oct. 2020, doi: 10.1007/s12652-020-01698-5.
[14] A. Anurag, R. Priyadarshi, A. Goel, and B. Gupta, "2-D Coverage Optimization In WSN Using A Novel Variant Of Particle Swarm Optimisation," in 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), 2020, pp. 663–668. doi: 10.1109/SPIN48934.2020.9070978.
[15] K. Zaimen, M. E. A. Brahmia, L. Moalic, A. Abouaissa, and L. Idoumghar, "A Survey of Artificial Intelligence Based WSNs Deployment Techniques and Related Objectives Modeling," IEEE Access, vol. 10, pp. 113294–113329, 2022, doi: 10.1109/ACCESS.2022.3217200.
[16] R. Priyadarshi, B. Gupta, and A. Anurag, "Wireless Sensor Networks Deployment: A Result Oriented Analysis," Wirel Pers Commun, vol. 113, no. 2, pp. 843–866, Jul. 2020, doi: 10.1007/s11277-020-07255-9.
[17] Z. Miao, X. Yuan, F. Zhou, X. Qiu, Y. Song, and K. Chen, "Grey wolf optimizer with an enhanced hierarchy and its application to the wireless sensor network coverage optimization problem," Applied Soft Computing Journal, vol. 96, Nov. 2020, doi: 10.1016/j.asoc.2020.106602.
[18] A. Chowdhury and D. De, "Energy-efficient coverage optimization in wireless sensor networks based on Voronoi-Glowworm Swarm Optimization-K-means algorithm," Ad Hoc Networks, vol. 122, Nov. 2021, doi: 10.1016/j.adhoc.2021.102660.
[19] S. Nematzadeh, M. Torkamanian-Afshar, A. Seyyedabbasi, and F. Kiani, "Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: an efficient metaheuristic-based method for environment-aware node deployment," Neural Comput. Appl., vol. 35, no. 1, pp. 611–641, Sep. 2022, doi: 10.1007/s00521-022-07786-1.
[20] I. A. Hasan and Osama A. Awad, "AN OPTIMIZED FUZZY LOGIC CONTROLLER FOR WIRELESS NETWORK CONTROL SYSTEM USING PSO," Iraqi Journal of Information and Communication Technology, vol. 5, no. 1, pp. 1–15, Apr. 2022, doi: 10.31987/ijict.5.1.180.
[21] S. Mirjalili, "The Ant Lion Optimizer," Advances in Engineering Software, vol. 83, pp. 80–98, 2015, doi: https://doi.org/10.1016/j.advengsoft.2015.01.010.
[22] A. A. Rustum and O. A. Awad, "LIFETIME MAXIMIZATION OF A MOBILE WSN USING ZRP-FUZZY CLUSTERING PROTOCOL BASED ON ANT-LION OPTIMIZER", Iraqi Journal of ICT, vol. 1, no. 1, pp. 70–82, Dec. 2021, doi: 10.31987/ijict.1.1.171.
[23] W. Chen, P. Yang, W. Zhao, and L. Wei, "Improved Ant Lion Optimizer for Coverage Optimization in Wireless Sensor Networks," Wirel Commun Mob Comput, vol. 2022, 2022, doi: 10.1155/2022/8808575.
[24] L. Abualigah, M. Shehab, M. Alshinwan, S. Mirjalili, and M. A. Elaziz, "Ant Lion Optimizer: A Comprehensive Survey of Its Variants and Applications," Archives of Computational Methods in Engineering, vol. 28, no. 3, pp. 1397–1416, May 2021, doi: 10.1007/s11831-020-09420-6.

