Maximising Coverage under Connectivity Constraint Utilising Nature-inspired Algorithms: A Comparative Analysis

Huy Nhat Minh Vu, Son Tran, Duc Manh Phan, Anh Hoang, Duc Chinh Hoang*
School of Electrical and Electronic Engineering
Hanoi University of Science and Technology, Hanoi, Vietnam
minh.vhn191973@sis.hust.edu.vn, son.t202691@sis.hust.edu.vn, manh.pd202452@sis.hust.edu.vn, anh.hoang@hust.edu.vn
*Corresponding author: chinh.hoangduc@hust.edu.vn

***Abstract*—Coverage and connectivity represent two of the most fundamental challenges in Wireless Sensor Networks stemming from their profound influence on network performance. This paper addresses these issues within the planned deployment context and presents meta-heuristic algorithms based on nature-inspired algorithms combined with graph theory to solve the problem. Four utilised algorithms are Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, and Artificial Bee Colony. Proposed methods are tested through Matlab simulations to verify the theoretical findings and practical effectiveness. The main contribution is how the initial population is meticulously generated, resulting in better performance to some extent in several case studies compared with other research findings, especially when applying the bee algorithm. Despite simplistic model assumptions, the methods still showcase the potential for practical application deployed in irregular shape areas, such as building environments with obstacles.**

***Keywords*—Node deployment, coverage and connectivity, metaheuristic algorithms, Wireless Sensor Networks**

I. Introduction

Wireless Sensor Networks (WSNs) play a pivotal role in emergent Internet of Things applications, such as smart buildings. WSN's primary usage is to perceive real-world environmental parameters [1]. The mechanism of WSN is a group of sensor nodes that collect physical parameters and transmit that sensed data to a designated node called the sink node. Typically, the connection between the sink node and sensor nodes is either single-hop or multi-hop communication, which depends on the communication protocol. Moreover, each sensor possesses two essential properties: sensing and communication range. A sensor covers a specific region if that region falls within its sensing range. Additionally, two sensors can communicate within each other's communication range. Understanding these properties is essential for network design considerations.

Node deployment is a fundamental aspect among design considerations, stemming from its direct impact on network metrics encompassing coverage, network connectivity, and energy efficiency [2]. From a methodology perspective, categories include random and deterministic deployment [3]. As the name suggests, the former deploys sensor nodes randomly, often resulting in a higher node than the optimal number across the monitored area. Thus, this technique is also referred to as dense deployment. Military applications or surveillance in hostile environments are in which this technique is chosen because of the area's inaccessibility. Typically, planes or aircraft drop sensors in the area of interest, and once deployed, the sensor's battery cannot be recharged. Therefore, the primary objective of these applications is to extend the entire network's lifetime by using energy efficiently. A viable solution is to develop a protocol to reduce the coverage overlap in the monitored area by controlling the active state of a sensor [4], [5].

On the other hand, deterministic or planned deployment refers to deliberately selecting optimised locations for sensor deployment. Applications having accessible monitored areas commonly employ this technique, such as intruder detection and environmental monitoring. The effectiveness of this method relies on certain assumptions, with common assumptions including the use of a two-dimensional area, the Boolean disk model and static, homogeneous properties. Typically, this approach aims to achieve multiple above metrics. We aim to apply the outcomes to smart building applications in which a viable power supply can be ensured for every node. Therefore, our primary focus is the coverage and connectivity problem (CCP).

CCP has been proven to be an NP-hard problem [6]; hence, two methods to solve the problem are exact and approximate techniques. The exact ensures finding the optimal solution, especially Integer Linear Programming, as in [7]–[9]. Reference [7] is the earliest to briefly formulate the mixed integer linear programming mathematical model about the CCP. Despite the primary focus being node scheduling to restrict energy usage, the paper defined the connectivity constraint clearly; that is, there is a path between an active sensor node and a sink node. Furthermore, the solution is acquired through the optimisation tool CPLEX 7.0, thus implying that such model complexity cannot be solved time-efficiently. Rebai proposes a simpler

This research is supported by the Vietnam Ministry of Public Security under the project number BCN.2022.T06.07.

979-8-3503-8106-1/23/$31.00 ©2023 IEEE

model that eliminates the energy constraint [8]. Based on Rebai's work, [9] presents two more mathematical formulations, resulting in better solutions and CPU computational time.

Obtaining optimal solutions demands extensive computation time; thus, the approximate method is favoured for large-scale problems. [8] employs the Genetic Algorithm (GA) and compares the results with pattern-based deployment regarding the sensor number to achieve coverage and connectivity. However, this algorithm exhibits inferior performance compared to [10] regarding objective functions and guaranteeing consistent valid chromosome generation. While these two studies address the coverage problem directly, treating connectivity as a constraint, [11] adopts a sequential approach. Firstly, the paper uses GA to solve the coverage problem and then introduces relay nodes to satisfy the connectivity constraints. Differing from the others, [12] considers deployment within an irregularly shaped area. Nevertheless, all the above studies exclusively utilise Genetic Algorithms.

In practical problems, preference leans toward approximate methods. However, to the best of our knowledge, a paucity of studies apply alternative meta-heuristic approaches to CCP and compare their performances. Therefore, this paper aims to address this gap by conducting a comparative analysis and selecting the most suitable approximation method for various case studies. This study proposes a meta-heuristic solution based on nature-inspired algorithms combined with graph theory. Nature-inspired algorithms, including Genetic Algorithm (GA) [13], Artificial Bee Colony Algorithm (ABC) [14], Ant Colony Optimization (ACO) [15], and Particle Swarm Optimization (PSO) [16], have been chosen based on [17]. In summary, the main contributions of this paper are:

- Proposed the method to generate the initial population for meta-heuristic approximation methods,
- Compared the proposed meta-heuristic methods to find the appropriate algorithm in various case studies,
- Compared our proposed methods with other research findings [18], [19].

II. Methodology

This section presents the problem formulation and introduces the proposed method in this study. The foundation of our work is built upon [8], [12].

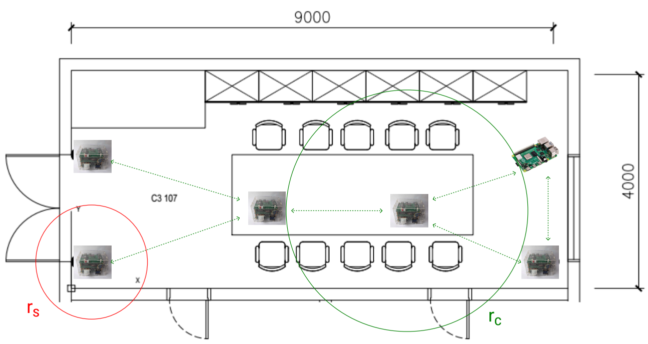


Fig. 1: An example of our WSN deployment in a building.

A. Preliminaries

The monitored area, denoted as A, is a two-dimensional rectangular plane divided into $x \times y$ identical points. The coordinate of a point is represented by (x_i, y_i). A group of sensor nodes $S = s_i \mid i = \overline{1,N}$ is deployed across the area at these specific points. Each sensor node possesses a sensing radius r_s and a communication radius r_c. The Euclidean distance between node s_i and node s_j is denoted as $d(s_i,s_j)$. Several assumptions are introduced to simplify the problem:

- The network uses the Boolean disk sensing model. A sensor node S covers a point $p(x, y)$ if its distance is less than that sensor sensing range:

$$P(s,p) = \begin{cases} 1, & d(s,p) \le r_s, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

- The connectivity is defined as at least one path from each sensor node to the sink node.
- Two sensors s_i and s_j are connected if the distance between them is less than or equal to r_c: $d(s_i, s_j) \le r_s$.
- All nodes are static, homogeneous and synchronised.

Before the mathematical model, the following definitions are established:

- The area covered by the sensor group S is defined as:

$$Area(S) = \sum_{i=1}^{N} \sum_{j=1}^{x \times y} P(s_i, p_j) \tag{2}$$

- A set C_i is defined as the set of neighbour sensors connected with the i-th sensor

$$C_i = \{j = \overline{1,N}, j \ne i \mid d(s_i, s_j) \le r_s\} \tag{3}$$

- A decision variable c is defined as:

$$c_i^k = \begin{cases} 1, & \text{if the i-th sensor is connected to the} \\ & \text{sink via k sensors, } k = \overline{1,N} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

B. Mathematical formulation

The paper aims to maximise the coverage area while ensuring network connectivity using the group of sensors. Thus, the fitness function is formulated as follows:

- Maximise the coverage area:

$$\text{Maximise } \frac{Area(S)}{A} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{x \times y} P(s_i, p_j)}{x \times y} \tag{5}$$

- Subject to the connectivity constraint:

$$\text{Subject to } c_i^k \le \sum c_j^{k-1} \text{where } k = \overline{1,N}, \ j \in C_i \tag{6}$$

C. Proposed solution

The generic flowchart of meta-heuristic algorithms follows the below steps. Each step is modified to solve the stated problem.

Step 1: Initialise the first population

Every individual (a chromosome, a bee, a particle, and an ant) within the population represents a potential solution. This study encodes an individual as a string of sensor coordinates (x_i, y_i). Consequently, the length of a solution is double the number of deployed sensors. For a collection of N sensors, an individual is represented as follows:

$$x_1\ y_1\ x_2\ y_2\ \ldots\ x_N\ y_N$$

In contrast to other studies employing the meta-heuristic method with randomly generated initial populations, our approach meticulously constructs the first generation, which is our pivotal contribution. **Algorithm 1** guarantees the connectivity constraint in the initial population by serialising each sensor position based on the specified conditions of the previously deployed node positions.

Algorithm 1: Initialize first population
Input: Sensors number N, Communication radius r_c, Area $(x \times y)$, Position of sink node (x_1, y_1)
Output: Full-connected first population

```
initPop = [ ]
Add sink node (x_1, y_1) to initPop
for i = 2 to N do
    Deploy node i(x_i, y_i) ensures 1 condition in:
        sqrt((x_i - x_1)^2 + (y_i - y_1)^2) <= r_c,
        ...
        sqrt((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2) <= r_c,
    Add node i (x_i, y_i) to initPop
end for
```

Step 2: Evaluate the fitness function

Equation (5) employs **Algorithm 2** to determine the coverage region. This algorithm divides the area into $x \times y$ points and calculates the coverage ratio by counting the points covered by sensor nodes, then dividing by the total points in the grid. Each individual in the population is assessed to identify the one with the most coverage area.

Algorithm 2: Compute coverage ratio of S
Input: Area size $(x \times y)$, Group of sensor nodes S
Output: Coverage ratio

```
Divide the area (x × y) to M points, create zeros
matrix C_{M×1}
for i = 1 to M do
    for j =1 to S do
        if distance (position(j),position(i))≤ r_s
            C_{i1} = 1 and break
    end for
end for
Coverage Ratio = (number of 1 in C) / M
```

Step 3: Meet termination conditions

After finding the optimal solution, two termination conditions are examined to determine whether the algorithm should stop or continue. Two criteria are defined: firstly, ensuring that the maximum run time remains below 15 minutes and, secondly, reaching the maximum number of iterations. If neither conditions are met, the algorithm continues with step 4.

Step 4: Generating new population

Generating the new individual follows the conventional idea of applied algorithms, GA [13], ABC [14], ACO [15], and PSO [16]. When a new member of the population is created, a connectivity constraint is considered. Inspired by (6), **Algorithm 3** uses graph theory to check the connectivity repeatedly in the generated member until that member is connected. After this step, the algorithm goes back to Step 2.

Algorithm 3 : Check connectivity constraints in each new population generated S
Input: New positions of N sensor nodes S
Output: Return true if connectivity, otherwise return false

```
Generate new population
Convert new population to adjacency matrix
C_{N×N}
        C_{ij} = 1 if d(s_i, s_j) ≤ r_c then
Convert the adjacency matrix to graph G
Compute connected components in graph G
if (number_connected_components = 1) then
    return true
else return false
```

For example, the idea based on the ABC algorithm is illustrated. This algorithm operates with a colony of artificial bees consisting of three groups of bees: employed, onlookers, and scouts bee. The colony is evenly divided, with the first half being the employed bees and the rest being the onlooker's bees. Firstly, initial solutions are generated, and only one employed bee exists for each solution. An employed bee modifies the solution in its memory depending on the local information, tests the fitness value, and ensures the connectivity constraint. If the fitness value of the new solution is higher than the old one, that bee memorises the new one and forgets the old solution. Once all employed bees complete this process, they communicate the solution information to the onlooker bees. The onlooker bees then evaluate information from all employed bees and select the solution based on probability, typically Roulette Wheel selection. Subsequently, the onlooker bee modifies that solution and updates if it is better. If a solution fails to improve through fixed cycles, that solution is abandoned. The abandoned solution is replaced with a new solution in the scout phase. The ABC algorithm simulates this process by randomly producing and replacing the abandoned location.

III. Results and Discussion

The following case studies were simulated using MATLAB x64 to assess the performance of the proposed algorithms. The results are then compared to other research algorithms under various experimental conditions. Additionally, a simulation of network coverage in an irregular area is conducted to demonstrate the algorithm's practical flexibility.

A. Case study 1: Examine the coverage area and the number of nodes.

This case study investigates the maximum coverage achieved through optimised sensor placement. Four algorithms are successively employed with parameters detailed in Table I. The population size and the number of iterations are defined considering each algorithm's coverage ratio and convergence speed during experimentation. The effects of node number variation are observed in Fig. 2, in which each point corresponds to the averaged result from ten simulation trials.

TABLE I: Case study 1 simulation Parameters

Parameter	Value
Selected area	100 m×100 m
Sensing range	10 m
Communication range	10 m
Number of sensors	10-60
Iterations	2000
Population size	50
Genetic Algorithm	
Crossover rate	0.7
Mutation rate	0.01
Particle Swarm Optimization	
Inertia Coefficient	1
Personal Acceleration Coefficient	1.5
Social Acceleration Coefficient	2
Ant Colony Optimization	
Pheromone Exponential Weight	1
Heuristic Exponential Weight	3
Evaporation Rate	0.2
Artificial Bee Colony	
Abandonment limit parameter	$100\times N$
Acceleration coefficient upper bound	1

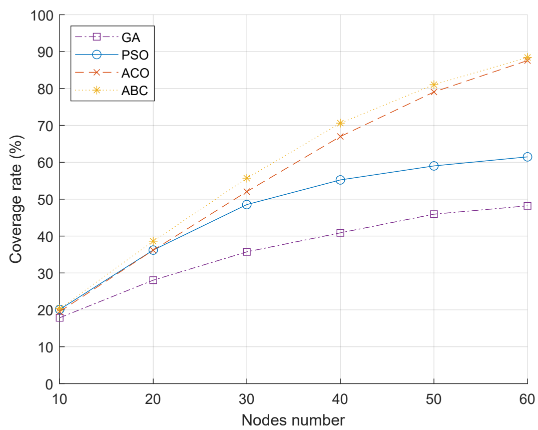


Fig. 2: Coverage rate comparison of proposed algorithms based on GA, PSO, ACO, and ABC

Fig. 2 illustrates a direct relationship between the number of nodes in the network and the achieved coverage percentage. ACO- and ABC-based algorithms demonstrate superior performance in the same number of nodes used compared to the GA- and PSO-based algorithms, especially starting from 30 nodes. While GA and PSO exhibit a similar trajectory, converging at an approximate coverage ratio of 50% and 70% with 60 nodes, respectively, the others show a steep and consistent ascent, reaching a peak coverage rate of approximately 90%.

Furthermore, algorithm robustness is investigated in Fig. 3. Firstly, the ten above results of each algorithm are normalized and then plotted to examine the mean, minimum, and maximum values. GA-based results scatter in the largest range, and the PSO-based algorithm has more outliers. Meanwhile, algorithms based on ACO and ABC show a higher stability than GA and PSO. Thus, the result implies that these two meta-

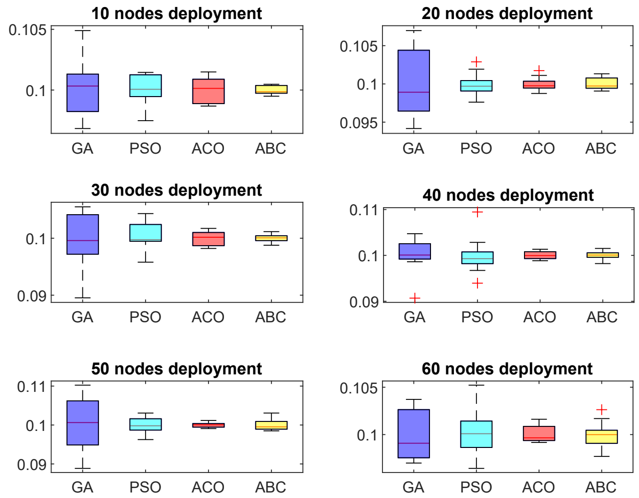


Fig. 3: Robustness comparison of proposed algorithms based on GA, PSO, ACO, and ABC

heuristic algorithms have a higher chance of finding optimal solutions. Hence, from now on, these two algorithms will be used to compare with other research findings.

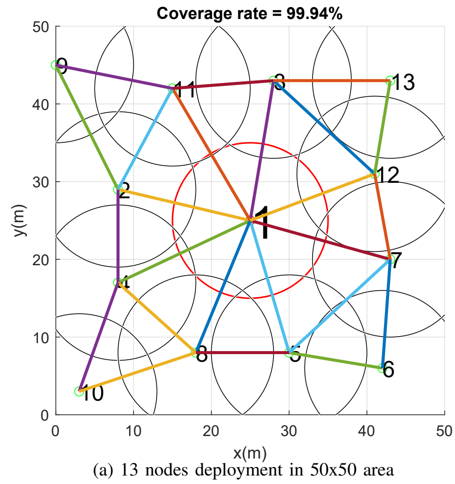
B. Case study 2: Achieve complete coverage and connectivity

TABLE II: Case study 2 simulation Parameters

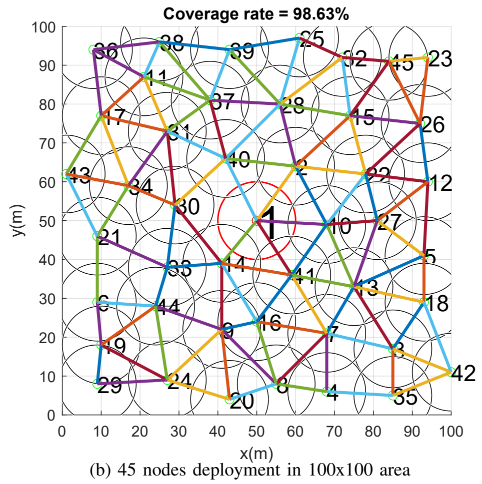
Parameter	Value
Selected area 1	50 m× 50 m
Selected area 2	100 m × 100 m
Sensing range	10 m
Communication range	20 m
Number of sensors	13, 45
Iterations	500
Population size	30

This case study changes the area size and communication range to find the minimum number of sensor nodes that achieve complete coverage and connectivity. Based on [18], the lower bound for the node number in the case of 50×50 and 100×100 area are 13 and 45, respectively. Fig. 4a and 5a demonstrate the former case in which the ACO-based algorithm attains a coverage rate of 98.06%, slightly lower than the 99.75% reported in ZainEldin's study. Conversely, our ABC-based solution surpasses this by 0.19%. Thus, adding just one more node could accomplish the case study objective.

In the latter scenario, the ABC-based algorithm maintains a performance level of 98.63% coverage in Fig. 4b. Meanwhile, our ACO-based solution in Fig. 5b yields comparatively inferior results contrasted with the ABC and the findings of [18]. This outcome also suggests that solutions based on GA and PSO fall short compared to the two-point crossover method employed in [18]. Nevertheless, our proposed solutions guarantee connectivity requirements in both cases, while ZainEldin's work depends on the number of sensors. Increasing the number of nodes is necessary to meet the case study's objective.

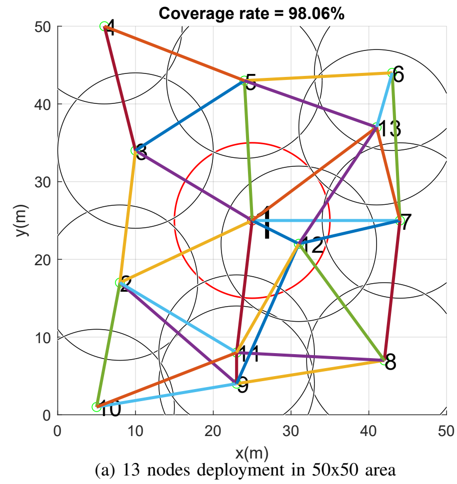


(a) 13 nodes deployment in 50x50 area

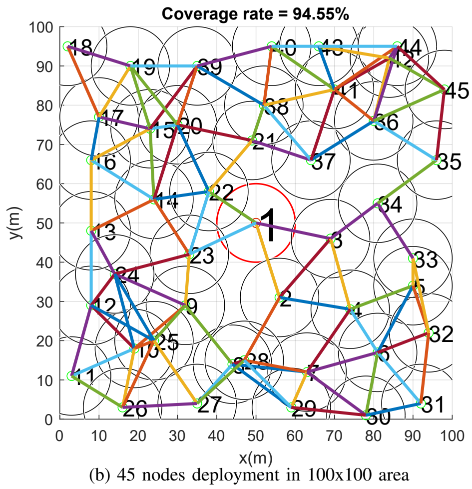


(b) 45 nodes deployment in 100x100 area

Fig. 4: ABC-based deployment



(a) 13 nodes deployment in 50x50 area



(b) 45 nodes deployment in 100x100 area

Fig. 5: ACO-based deployment

TABLE III: Case study 3 simulation Parameters

Parameter	Value
Selected area	200 m× 200 m
Sensing range	20 m
Communication range	20 m
Number of sensors	40, 50, 60
Iterations	3000
Population size	50

C. Case study 3: Investigate the convergence speed

Convergence speed is investigated in this case study. The selected area has been resized to 200m×200m, resulting in 40,401 available locations to deploy sensor nodes. The conventional idea of ACO to solve the travelling salesman problem is impractical in this large-scale problem because the pheromone matrix denotes switching from i to j position would have a size of 40401×40401; hence, this case study focuses solely on the ABC-based algorithm. Despite the ABC idea to modify the best solution, a higher number of iterations is necessary to address this large-scale problem, allowing to observe the convergence speed in Fig. 6. The parameters follow [19], table III. Even after 500 iterations, the fitness value of our proposed methods continues to increase steadily. Convergence of the

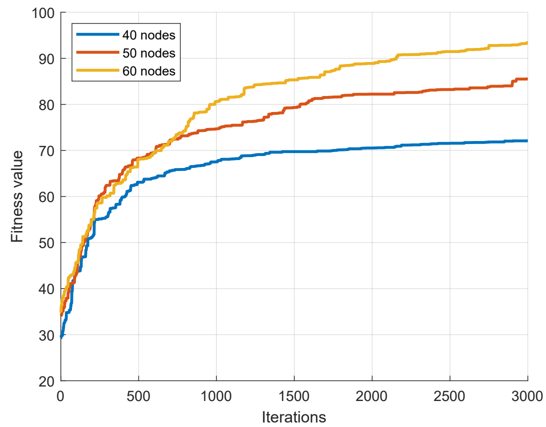


Fig. 6: ABC-based algorithm convergence speed

solution becomes evident at approximately 2,000 iterations. While our coverage ratio may not match that of [19], it is worth noting that Yue's work demands a significantly higher number of nodes, precisely over 120, to achieve full connectivity.

D. Case study 4: Deploy nodes in an irregular area

In this case, the monitored area shape is irregular, with a no-interest area in the centre. Firstly, covert the inspected

TABLE IV: Case study 4 simulation Parameters

Parameter	Value
Sensing range	10 m
Communication range	10 m
Number of sensors	100
Iterations	15000
Population size	50

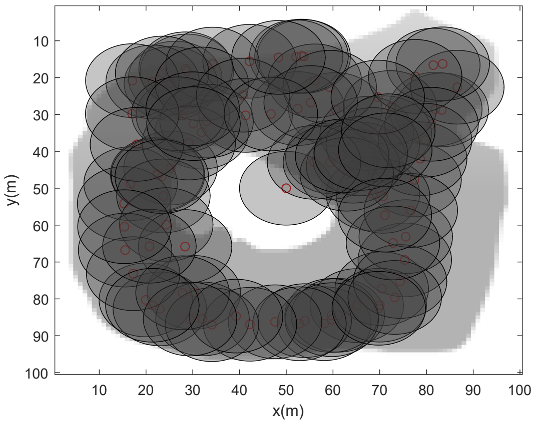


Fig. 7: Deployment in an irregular area.

area to a bitmap with the available location denoted as 1; nodes must be deployed in these marked positions. Then, the proposed algorithm utilised ABC is applied. The sink node is located in the centre of the map, and the sensor nodes are deployed in available locations, avoiding the no-interest area. Fig. 7 showcases the algorithm's potential deployment in an irregular area even though a significant number of nodes and iterations are required. This case study suggests that the proposed algorithms could maximise area coverage with a given number of sensors in architectural planning for buildings. Furthermore, the results might aid in determining the optimal number of sensors required to monitor the entire area.

IV. Conclusion

This paper investigates deterministic node deployment to achieve the maximum coverage area and complete connectivity. Four meta-heuristic algorithms based on nature-inspired algorithms combined with graph theory are compared through intense simulation in various case studies, including ABC, ACO, PSO, and GA. The pivotal contribution is how the first population is generated, leading to higher coverage and consistently achieving complete connectivity in some cases compared with other research findings, especially when employing the bee algorithm. Additionally, the proposed method demonstrates its potential applicability in irregularly shaped environments, such as building environments with restricted availability. Nevertheless, the study is undermined by simplistic assumptions, including using a two-dimensional monitored area and the binary disk model. Thus, the probabilistic model should be employed in the future. Furthermore, the proposed method will be expanded to tackle multi-objective optimization problems, including energy savings and network lifetime metrics.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms, and Open Problems," IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 550–586, 2017.

[3] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," Ad Hoc Networks, vol. 6, no. 4, pp. 621–655, 2008.

[4] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in Proceedings of the 1st international conference on Embedded networked sensor systems, in SenSys '03. New York, NY, USA: Association for Computing Machinery, pp. 28–39, Nov. 2003.

[5] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," Ad-Hoc and Sensor Wireless Networks, vol. 1, no. 1–2, pp. 89–124, 2005.

[6] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," Pervasive and Mobile Computing, vol. 4, no. 3, pp. 303–334, Jun. 2008.

[7] F. G. Nakamura, F. P. Quintão, G. C. Menezes, and G. R. Mateus, "An Optimal Node Scheduling for Flat Wireless Sensor Networks," in Networking - ICN 2005, P. Lorenz and P. Dini, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 475–482, 2005.

[8] M. Rebai, M. Le berre, H. Snoussi, F. Hnaien, and L. Khoukhi, "Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks," Computers & Operations Research, vol. 59, pp. 11–21, Jul. 2015.

[9] S. Elloumi, O. Hudry, E. Marie, A. Martin, A. Plateau, and S. Rovedakis, "Optimization of wireless sensor networks deployment with coverage and connectivity constraints," Ann Oper Res, vol. 298, no. 1, pp. 183–206, Mar. 2021.

[10] S. K. Gupta, P. Kuila, and P. K. Jana, "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks," Computers & Electrical Engineering, vol. 56, pp. 544–556, Nov. 2016.

[11] N. T. Hanh, H. T. T. Binh, V. Q. Truong, N. P. Tan, and H. C. Phap, "Node placement optimization under Q-Coverage and Q-Connectivity constraints in wireless sensor networks," Journal of Network and Computer Applications, vol. 212, p. 103578, Mar. 2023.

[12] F. Tossa, W. Abdou, K. Ansari, E. C. Ezin, and P. Gouton, "Area Coverage Maximization under Connectivity Constraint in Wireless Sensor Networks," Sensors, vol. 22, no. 5, Art. no. 5, Jan. 2022.

[13] "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence — MIT Press eBooks — IEEE Xplore." https://ieeexplore.ieee.org/book/6267401 (accessed Sep. 07, 2023).

[14] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report - TR06," Technical Report, Erciyes University, Jan. 2005.

[15] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 26, no. 1, pp. 29–41, Feb. 1996.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks, vol.4, pp. 1942–1948, Nov. 1995.

[17] A. Singh, S. Sharma, and J. Singh, "Nature-inspired algorithms for Wireless Sensor Networks: A comprehensive survey," Computer Science Review, vol. 39, p. 100342, Feb. 2021.

[18] 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 Human Comput, vol. 11, no. 10, pp. 4177–4194, Oct. 2020.

[19] Y. Yue, L. Cao, and Z. Luo, "Hybrid Artificial Bee Colony Algorithm for Improving the Coverage and Connectivity of Wireless Sensor Networks," Wireless Pers Commun, vol. 108, no. 3, pp. 1719–1732, Oct. 2019.