Coverage maximization of sensor networks with connectivity constraints in obstacle-filled environment based on nature-inspired algorithms

Anh Tran Quang, Tuyen Pham Huy, Du Tran Phuong, Son Tran and Duc Chinh Hoang*

School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Vietnam
**Corresponding author E-mail:chinh.hoangduc@hust.edu.vn*
DOI: https://doi.org/10.64032/mca.v29i1.288

Abstract

The application of meta-heuristic algorithms has significant potential in various fields, including wireless sensor networks. In this paper, we utilize two algorithms, the Fruitfly optimization algorithm (FOA) and the Nutcracker optimization algorithm (NOA), to address two critical issues: optimizing coverage and ensuring connectivity in sensor networks. The main contribution of this paper is the application of these algorithms to arbitrary communication radius, independent of predefined connectivity assumptions. Simulation results demonstrate the effectiveness of the proposed methods by comparing with each other and with the two traditional algorithms Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Additionally, this paper simulates the coverage area in an environment with different types of obstacles to showcase the practical flexibility of the algorithms.

Keywords: *Wireless Sensor Networks; Node deployment; CC-CM problem; Metaheuristic algorithms.*

Symbols

Symbols	Units	Description
A		set of monitoring points
B		set of obstacles
S		set of sensor nodes
r_c	m	communication radius
r_s	m	sensing radius

Abbreviations

WSNs	Wireless Sensor Networks
CC-CM	Connectivity Constrained - Coverage Maximization
NP-hard Problem	Non-deterministic Polynomimal time hard problem
NOA	Nutcracker Optimization Algorithm
FOA	Fruitfly Optimization Algorithm

1. Introduction

Wireless Sensor Networks (WSNs) can be considered a distributed system, which contain a large number of sensor nodes that have abilities of monitoring data and communicating wirelessly. The idea is to use those nodes to collect environmental data, then transfer them to a special node (called Base station or Sink node) for further analyzing and decision-making. Thanks to the flexible architecture of WSNs, it can be used in a considerable range of applications such as Internet of Things[1], Smart building monitoring[2] and Military purposes [3],...

In the operation of WSNs, one of the main challenges is the deployment of sensor nodes. To evaluate the effectiveness of node localization, some common metrics are introduced, which can be listed as coverage, energy consumption, network lifetime and connectivity [4]. This paper focuses on the work of maximizing the coverage ratio while ensuring connectivity constraint, which is categorized as the Connectivity Constrained - Coverage Maximization problem (CC-CM) [5]. The CC-CM is a Non-deterministic Polynominal time (NP-hard) problem [4]. This type of problem is computationally and chronically expensive to find a global optimal solution. Popular approaches for NP-hard problems are meta-heuristic algorithms [6] as they are more flexible in searching the global solution field with less time and resources.

In the perspective of solving the Coverage Maximization problem only, number of researches have been published making use of the supreme performance of meta-heuristic algorithms. In [7], Jianghao proposes the Yin-Yang Pigeon-inspired optimization algorithm to maximize coverage ratio of a sensor network in a rectangular area, results show that the algorithm works well with simple scenarios but do not mention the problem of more complex environments. Zhendong Wang et al.[8] applies the improved Grey wolf optimizer to solve node coverage maximization with obstacles in a rectangular area. The authors design a new nonlinear convergence factor instead of the old linear one to balance the global and local search.

In some researches, both Connectivity constraint and Coverage Maximization problems are considered, making the CC-CM problem. In [9], an algorithm based on the Genetic Algorithm, called IDDT-GA is proposed to maximize the coverage ratio

Received: 12 March 2025; Revised: 28 April 2025; Accepted: 28 April 2025.

while lowering the number of sensor nodes. Results show that the use of IDDT-GA comes out with superior solutions compared to other algorithms such as Immune Algorithm (IA), Harmony Search (HS) and Whale Optimization Algorithm (WOA). On the other hand, Nematzadeh [10] suggests utilizing the improved version of Grey Wolf Optimization named as Mutant-GWO. The research has focused on the work of creating a network topology while maximizing coverage of custom areas. However, to the best of our knowledge, most of the researches solving the problem of connectivity constraint make use of a condition of communication range which mentioned in [11]. This condition helps simplify this problem to the coverage maximization problem as [11] proves that connectivity will always be ensured if the covered region is convex and the communication range is more than twice of sensing range.
The key contributions of this paper are as follow:

- This paper introduces the task of solving the CC-CM problem with any value of ratio between communication range and sensing range.
- Two nature-inspired algorithms are adopted and modified to maximize the coverage ratio and maintain the connectivity constraint simultaneously. After that, their performances are compared to other classical algorithms.
- Deployment simulations are firstly held on a basic plain rectangular space, after that, some custom areas are selected for node placement in order to serve further practical purposes.

The structure of this paper is organized as: Section 1 presents introduction to the CC-CM problem and literature survey. Section 2 constructs the mathematical formulation of the system model and the two metaheuristic algorithms. Section 3 illustrates simulation results and discussion of the algorithms' performance. Final section shows the conclusion and future works.

2. Methodology

This section firstly introduces formulation of the network model and the two main problems of coverage and connectivity. Secondly, two meta-heuristic solutions Nutcracker and Fruitfly are studied to enhance theirs performance on solving the CC-CM problem. The network model in this paper is built upon the foundation established in Reference [14] while the two meta-heuristic algorithms are grounded in the principles outlined in [12] and [13].

2.1. Mathematical formulation

2.1.1. System model

The objective of this paper is to find a node deployment that maximize coverage ratio while ensuring connectivity within a monitoring region with a condition of any communication range values, as previously mentioned. The monitoring area is a two-dimensional rectangular space divided into a finite set of points (A) with $X \times Y$ coordinates. The set of obstacles is denoted as $B = \{b_n \mid n = \overline{1,OB}\}$, where each element b_n (x_n, y_n) is a point in total of OB points representing the location of an obstacle. A set of sensor nodes $S = \{s_n \mid n = \overline{1,N}\}$ is deployed across the area which represents the solution vector. Each sensor has a communication radius r_c and a sensing radius r_s. The Euclidean norm denoted as $\|\cdot\|$. Our assumptions are as follows:

- All sensors are homogeneous and static.
- The network uses the binary Boolean disk sensor model.
- A point k_i is covered by a sensor s_j if $\|s_j - k_i\| \leq r_s$.
- The fully connected constraint is defined as there is at least one path for every sensor node to transfer data to sink node.
- Sensors s_l are counted as connected with s_m if $\|s_l - s_m\| \leq r_c$.

Firstly, the following definitions are established:

- The space covered by the sensor group S is defined as a set of points that are in at least one sensor's coverage zone and are not in obstacle set B:

$$\text{Area}(S) = \sum_{n=1}^{X\times Y} \min\left(1, \sum_{m=1}^{N} K(s_m, k_n)\right)$$
$$\text{where } K(s,k) = \begin{cases} 1, & \text{if } \|s-k\| \leq r_s \text{ and } k \notin B, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

- A set L_i is defined as set of local sensors connected with the i-th sensor:

$$L_i = \{j = \overline{1,N},\ \|s_i - s_j\| \leq r_c,\ j \neq i\} \tag{2}$$

- A decision variable c is defined as:

$$c_n^k = \begin{cases} 1, & \text{if the } n\text{-th sensor connects to the} \\ & \text{sink via } k \text{ other sensors, } k = \overline{1,N} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The paper aims to maximize the coverage ratio in the unobstructed space while ensuring full connectivity. Hence, the fitness function is formulated as:

- Maximize the coverage ratio of the area of interest:

$$\text{Maximize} \quad Cov(S) = \frac{\text{Area}(S)}{X \times Y - OB} \tag{4}$$

- Subject to the connectivity constraint:

$$c_n^k \leq \sum c_m^{k-1} \quad \text{where } k = \overline{1,N}, \quad m \in L_n \tag{5}$$

Further explanations of the fitness function and the constraint are demonstrated in following Fig. 1 and Fig. 2:

In Fig. 1, a deployment of 6 nodes in a 100 × 100 m^2 environment with a rectangle obstacle is presented. The green points represent interest area that is covered by the sensor network while the red points represent for the obstacle area that are covered by the network. As the constraint of obstacle avoiding has not been applied, there is a node that stuck inside the obstacle. Following the formula 4, the coverage can be calculated as taking the ratio between the green points and the white area points.
Figure 2 illustrates the two samples of a unconnected and connected network. Formula 5 is implemented using Graph Theory through the number of connected components in the network's Graph. In Fig. 2b, the network maintains connectivity as all the six nodes contribute to one connected

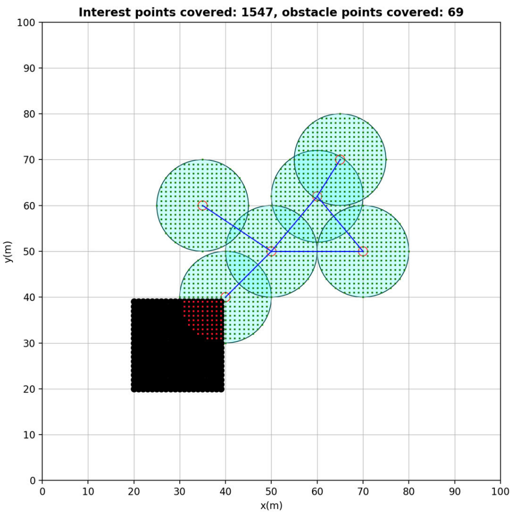


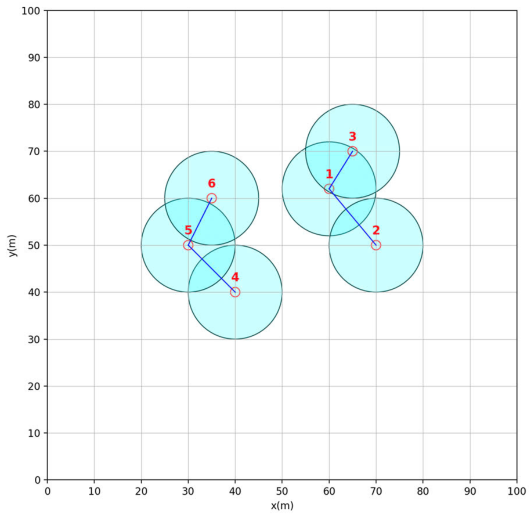
Figure 1: Coverage Ratio Calculation

component, leading to the fact that always exist at least a path from every node to the Node 1. On the other hand, Fig. 2a shows a case that the network loses its connectivity as the six nodes generate two connected components making node 4, 5 and 6 do not have any paths to communicate with the node 1, 2 and 3.

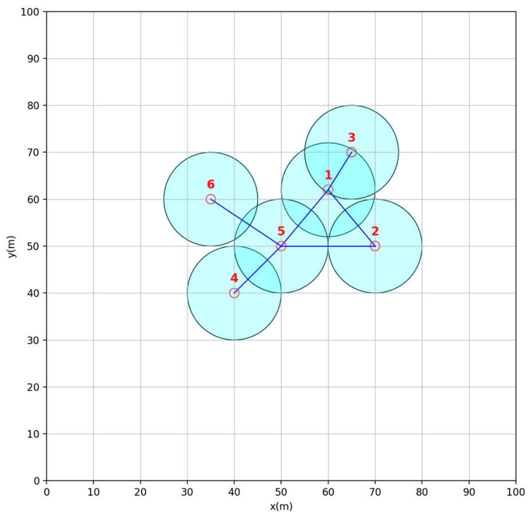
2.1.2. Connectivity Constraint & Coverage Maximization

The CC-CM problem focuses on finding a combination of sensor locations such that those sensor nodes form a network that satisfies the connectivity condition while maximizing its coverage. Initialization is the first step to establish the first populations of solution. This paper uses a constrained stochastic initialization method where sensors' positions are randomly generated in a small surrounding space of the sink node which easily ensuring connectivity constraint and avoiding any overlap with obstacles.

After that, the initial solutions need to be optimized to improve overall coverage ratio and minimize overlapping or uncovered areas. Metaheuristic algorithms such as Nutcracker optimization and Fruitfly optimization are chosen to identify optimal positions for the sensor group to maximize the fitness function. These algorithms simulate nature-inspired processes and can gradually improve the sensor positions to achieve better coverage. At every step of adjusting the positions of the sensors, it is compulsory to ensure that connectivity is maintained and obstacles are avoided. This is achieved by restricting the movement of the sensors within the communication radius r_c from neighboring sensors and performing thorough check at each iteration of the optimization phase. With this approach, the system not only optimizes coverage effectively and ensures connectivity but also maintains feasibility and stability in environments with complex obstructions.



(a) Unconnected network



(b) Connected network

Figure 2: Connectivity constraint explanation

2.2. Proposed solutions

2.2.1. Nutcracker optimization algorithm

Clark's Nutcrackers are solitary birds with grey pale and black wings. Their main food source is pine seeds. [12] They prefer to choose top-quality seeds because of the bigger size and easier harvesting, and they would tend to collect them on the tree with a large cone density. They store pine seeds in numerous storages in the autumn, then they will use their spatial memory to recall these locations in the winter. The Nutcracker Optimization Algorithm (NOA) is inspired by their intelligence, which involves two key phases: (1) The foraging and storage strategy and (2) The cache-search and recovery strategy.

In the first strategy, nutcrackers search for seeds at pine cones, each nutcracker inspects the potential locations. If the food is not promising, they will seek another cone in another position within pine trees or other trees. This behaviour is defined as the first exploration phase. Otherwise, they will transfer the food to caches, which is called the first exploitation phase. In the second strategy, nutcrackers use spatial memory and nearby cues to retrieve caches during winter. Two reference points (RPs) are defined to mimic this behavior, in which the 1^{st} RP refines the current position for local exploration, while the 2^{nd} RP expands the search space to explore new regions for hidden caches. This behaviour is defined as the second exploration phase. A clear explanation of the algorithm is demonstrated in **Algorithm 1**.

The main idea of the purposed solution inspired by NOA is utilizing the searching capacity with high convergence speed and escape ability from the local optimum to reach the near-optimal solution. The operation of NOA relies on three main parameters: P_{a_1} for adjusting the rate of updates to the current solution in the unexplored areas for covering intractable regions, P_{a_2} for determining the switch between cache-search and recovery stages, and δ for considering if the current solution will be updated within the upper and lower bound of the optimization problem to control the ability of avoiding local optima [12]. The smaller of δ, the more effort of searching the solutions globally in the problem instead of exploring around a specific solution.

In this paper, the purposed NOA is adjusted to adapt the problem scale and to utilize the searching capacity with high convergence speed and escape ability from local optimum to reach the near-optimal solution. Specifically, the created reference points have to satisfy the obstacles constraint. The position of agents updated by using reference points will be done successfully after achieving the satisfaction of connectivity condition.

Algorithm 1: Nutcracker Optimization Algorithm
Input: Parameter A, B, N, r_c, r_s, MaxIt, nPop, P_{a_1}, P_{a_2}, δ
Output: Best solution and best fitness

```
Initiate nPop nutcrackers S^1, S^2, ...S^nPop, which S^j = {s_i^j | i = 1,N}
for t = 1 : MaxIt do
    for j = 1 : nPop do
        Generate σ, σ1, ϕ, φ in the range of [0,1]
        if σ < σ1: /* First strategy */
            if ϕ < P_a1: /* Exploration phase 1 */
                Update S^j by searching food behaviour using δ
            else: /* Exploitation phase 1 */
                Update S^j by caching behaviour
            end if
        else: /* Second strategy */
            Generating RP matrix
            if φ < P_a2: /* Exploration phase 2 */
                Update S^j by using spatial memory
            else: /* Exploitation phase 2 */
                Update S^j by recovering cache behaviour
            end if
        end if
    end for
    Update best solution and best coverage rate
end for
```

2.2.2. Fruitfly optimization algorithm

The Fruitfly Optimization Algorithm (FOA) [13] is inspired by nature, based on the foraging behavior of fruitfly. The characteristic of fruitflies is their ability to quickly detect food sources due to their superior sense of smell and vision compared to other species. First, they use their sense of smell to gather scents from the air to determine the direction of the food source. Once the direction is established, they move towards the food source and they get closer, they use their vision to accurately approach the food source and begin to exploit it.

Detailed steps for implementing the FOA in CC-CM problem are as followen in **Algorithm 2**.

Algorithm 2: Fruitfly Optimization Algorithm
Input: Parameter A, B, N, r_c, r_s, MaxIt, nPop, step size
Output: Best solution and best fitness

```
Setup Parameters , initiate nPop flies S^1, S^2, ...S^nPop, each fly is a solution containing the initial positions of the nodes S^j = {s_i^j | i = 1,N}
for It = 1 to MaxIt do
    for i =1 to nPop do
        A position of node i in solution j-th is randomly chosen: s_i^j
        s_i^j' = s_i^j +(2*step*rand(0,1) - step).
        if S^j' satisfies connectivity constraint:
            if Cov(S^j') > Cov(S^j)
                S^j = S^j'
            end if
        end if
    end for
    Update best solution and best coverage rate
end for
```

A large step size allows the algorithm to quickly cover the search space, explore more potential regions and converge faster. However, its local search ability is quite weak. A small step size allows the algorithm to explore the surrounding area of a region, allowing for more precise adjustments and potentially leading to better results. However, its global search ability is weak, it may get stuck in a local optimum and the convergence speed is reduced. It is necessary to choose a step size that is appropriate to the problem. Each node will be adjusted with a small amount of variation, rand(0,1) is a random number between 0 and 1; thus, the range of the additional values is in [-step, step]. In this situation, a step size of 30 is selected because the maximum value of it allows nodes to move to the further corner of the map, which reduces the number of iterations required and the possibility of getting stuck in a local optima.

3. Results and Discussion

All simulation studies are conducted in Python environment on an Ubuntu Computer equipped with Intel Ultra 155H CPU and 32GB RAM. Parameters used for network and algorithms simulation are clearly shown in Table 1. In case study 1, the monitoring area is a plain rectangle region, this focuses on the basic cases of changing the number of nodes and the communication radius to observe the performance of the two algorithms FOA and NOA comparing to two well-known algorithm GA and PSO under different circumstances of communication ranges. In case study 2, simulations of 60 nodes deployment on 2 different regions, firstly on an area with a rectangular obstacle and secondly on the map layout of a part of Hanoi University of Science and Technology (HUST) campus in Vietnam, are presented to verify the practical flexibility of the two algorithms when deploying in real environment.

3.1. Case study 1

Firstly, this case study emphasizes the superior advantage of this paper's propose solution that its deployment solutions do not depend on condition of the ratio between communication range and sensing range. This means that the algorithms can come out with solutions that will fully maintain the

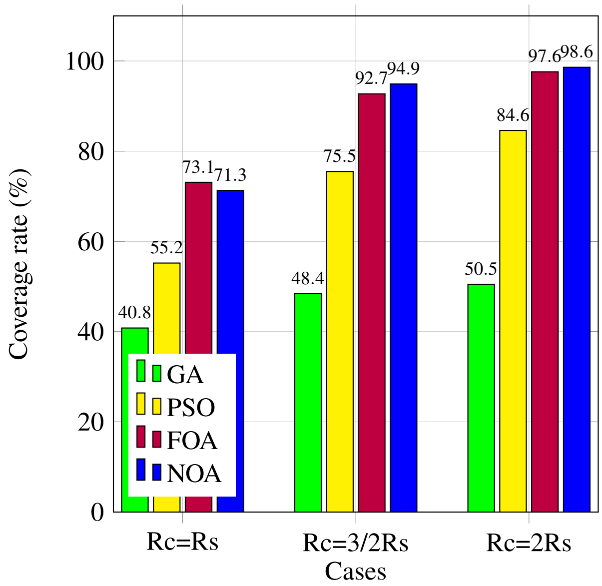
Table 1: The network and algorithms parameters

Network Model	
Parameter	**Value**
Selected area (A)	100 m × 100 m
Sensing range (r_s)	10 m
Communication range (r_c)	10, 15, 20 m
Number of sensors (N)	40, 60
Iterations (MaxIt)	2000
Population size (nPop)	50
Fruitfly Optimization	
step size (step)	30
Nutcracker Optimization	
P_{a_1}	0.2
P_{a_2}	0.2
δ	0.05
Genetic Agorithm	
Crossover rate	0.7
Mutation rate	0.01
Particle Swarm Optimization	
Inertia Coefficient	1
Personal Acceleration Coefficient	1.5
Social Acceleration Coefficient	2

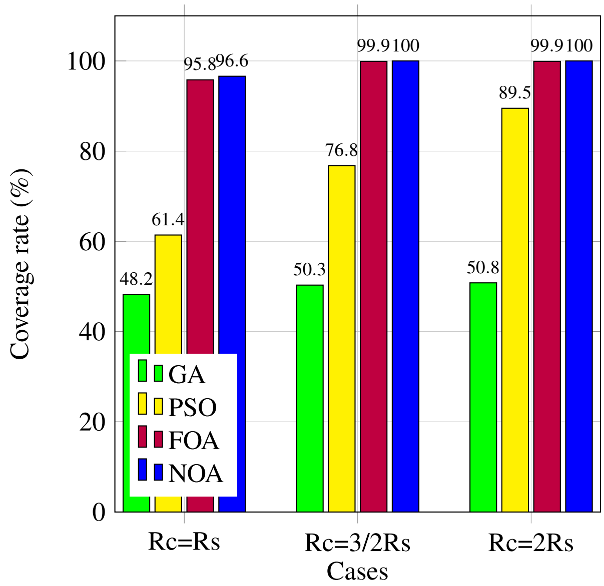
connectivity constraint as well as be capable of working with any values of communication radius filled in the network parameter.

Based on the chart in Fig. 3, it is noticeable that when the communication radius increases, the coverage rate also increases. In 40-nodes case, it can be claimed that the performances of NOA and FOA are relatively comparable in all cases and the results completely outperform those of the traditional algorithms GA and PSO. It can be easily seen that GA is not suitable in the use of constrained first population, as the crossover phase changes the position of the sensor node dramatically, which easily violate the connectivity constraint. Therefore, the results of GA do not change significantly when the scenario is modified. PSO, although yielding results inferior to the two algorithms NOA and FOA, still produces progressively better outcomes when relaxing the conditions on the ratio between r_c and r_s or increasing the number of sensors, demonstrating its suitability for the constrained deployment method. On the other hand, the coverage experiences a remarkable expand to 92.76% and 94.93% when the communication-sensing ratio increases from $r_c = r_s$ to $r_c = 1.5r_s$ using FOA and NOA, respectively. In addition, a small coverage improvement of about 5% in case of $r_c = 2r_s$ as most of the easy-to-reach area has been covered. In 60-nodes case, the coverage value obtained by NOA easily achieved 94.96% in $r_c = r_s$ case, and 100% coverage in other 2 cases, whereas the performance of FOA is slightly worse. Fig. 3 indicates that regarding various cases, both FOA and NOA significantly outperform the traditional algorithms and successfully resolve a CC-CM with a random ratio between communication range and sensing range.

To conclude, with the condition of $r_c = 2r_s$ both high coverage and connectivity conservation tasks are more effortless to achieve no matter of the algorithms used. Therefore, this paper solves a more complicated problem when successfully resolve a CC-CM with a random ratio value between communication range and sensing range.



(a) Deployment with 40 nodes



(b) Deployment with 60 nodes

Figure 3: Comparison of the coverage rates between GA, PSO, FOA and NOA in case of different numbers of nodes and communication-sensing radius ratio.

3.2. Case study 2

Fig. 4 and Fig. 5 illustrate the 40 and 60 nodes placement in the area containing a rectangular obstacle, which is located by two corner points (60, 80) and (80, 20). Initialized node positions tend to concentrate around the sink node, and they are distributed more evenly through the iterations. In the case of 40-node demonstrated in Fig. 4a and Fig. 4b, both of the two algorithms comes out with nearly 80% of coverage as the area in the side of obstacle-free is effectively covered. Performance of the two algorithms shows an area that is hard to cover which is behind the obstacle. Both of the algorithms fail to reach the other side of the obstacle as there is no sensor node cover the eastern side of the map. However, there are signs of FOA reaching into hard-to-reach areas as some of the nodes attempt to search that area in both southeast and northeast corners;

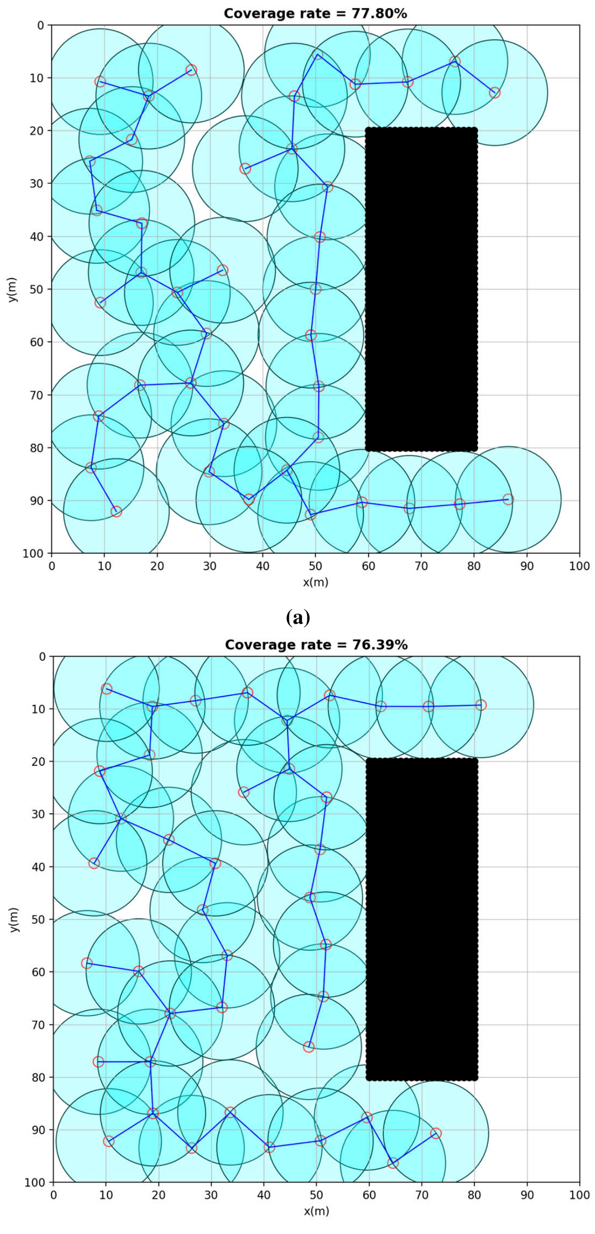


Figure 4: 40 nodes deployment in rectangular obstacle case using (a) FOA (b) NOA

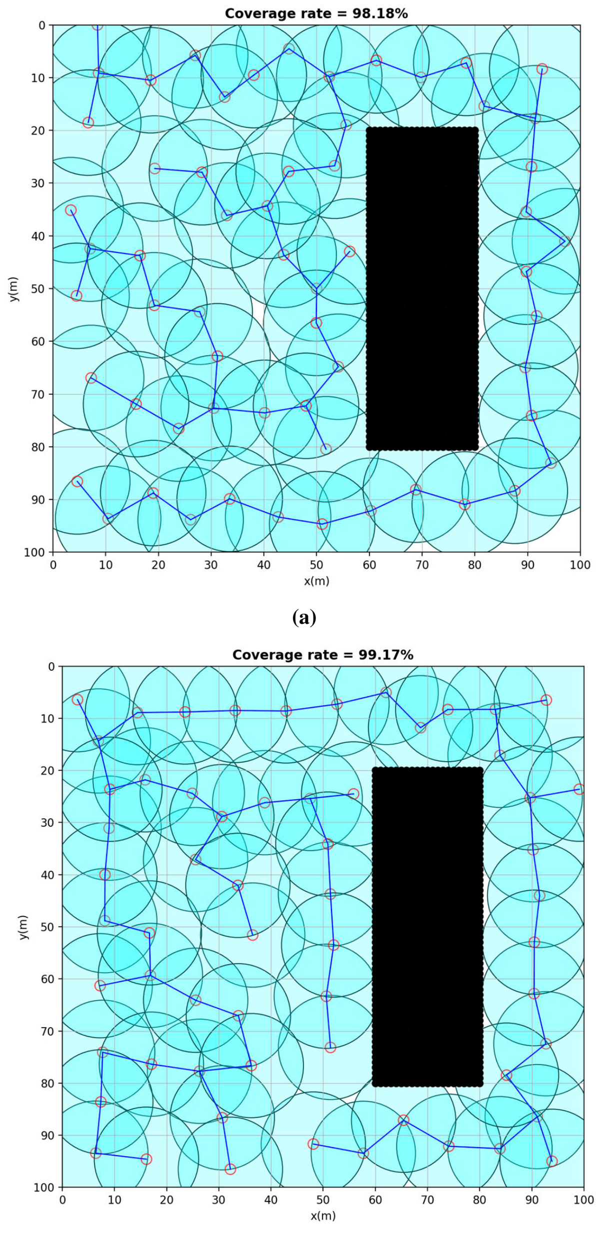


Figure 5: 60 nodes deployment in rectangular obstacle case using (a) FOA (b) NOA

however, due to the relatively small number of sensors, it is not sufficient to cover that part.

Furthermore, it can be easily seen that the farther the area from the initial source base, the more sparsely distributed the nodes are due to the random factors in the algorithm, and the degree of sparsity will be greater with the appearance of the obstacles. This behavior of FOA is clearly shown in Fig. 5b. Despite of considerably increasing the number of sensor nodes deployed, there is still a small area in the bottom right corner of the map is not covered. In contrast, demonstrated in Fig. 5a, NOA seems to distribute sensor nodes more evenly, resulting in a slightly higher coverage ratio of over 99% coverage. This indicates that the ability to avoid obstacles and the flexible searching ability of NOA in this case are better than FOA. The results obtained from FOA and NOA are relatively satisfactory and can be applied as the deployment is able to avoid the prohibited area while covering up to nearly 100% of the necessary space. However the case of a ordinary rectangle obstacle is not practical as real world cases performs highly fragmented or irregular obstacle layouts.

In order to deal with that problem, regarding from Fig. 6 and Fig. 7, a building terrain is used to examine the flexibility of the two algorithm in 40 and 60 node-deployment, respectively. The obstacle-filled environment taking from a real location showcasing the square area of C1 building in Hanoi University of Science and Technology is obtained by firstly filtering for locations of buildings, which is considered as obstacles, based on brightness thresholds. After that, the map is resized to 100×100 bitmap and encoded to a matrix map with value 0 representing the interest points and the value -1 represents the obstacle points that are moved to set B. In the case of more complicated obstacles, such as buildings in the terrain, the behavior of FOA and NOA in finding the best solution is more obvious. Fig. 6 shows the case of 40-node deployment generated by the two algorithms. It can be easily seen that the lack of nodes number leads to relatively low coverage ratio

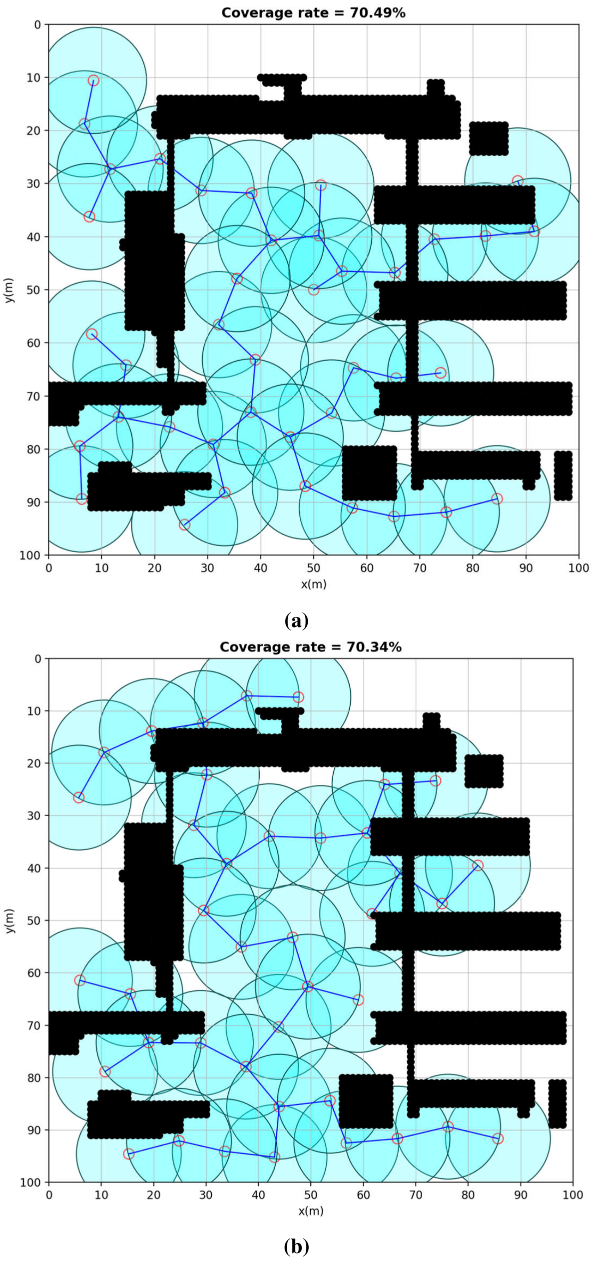


Figure 6: 40 nodes deployment in building obstacles case using (a) FOA (b) NOA

and the corner regions have not been accessed. However, the final coverage ratio of both algorithms still reach over 70% as the center of the terrain is nearly fully covered. Additionally, each algorithm exploits the southern space relatively well since that area is not blocked by a long building as in the north. On the other side, due to the randomness of the algorithm, FOA explores the northwest region of the map while NOA successfully covered half of the northern area located behind the long building.

Even though the nodes number increases to 60 nodes, in Fig. 7b, the inaccessible areas such as the area at 1 o'clock and 5 o'clock corners, which are isolated due to the surrounding obstacles, or the neighboring areas are not fully covered. Similarly, in the upper side of the map, there are several areas are significantly uncovered when deploying by NOA. However, the coverage rate is faintly higher with respect to FOA in Fig. 7a. Another point that should be noticed is that the number of uncovered regions in FOA is less than NOA and they are often concentrated in hard-to-reach areas, however, the area

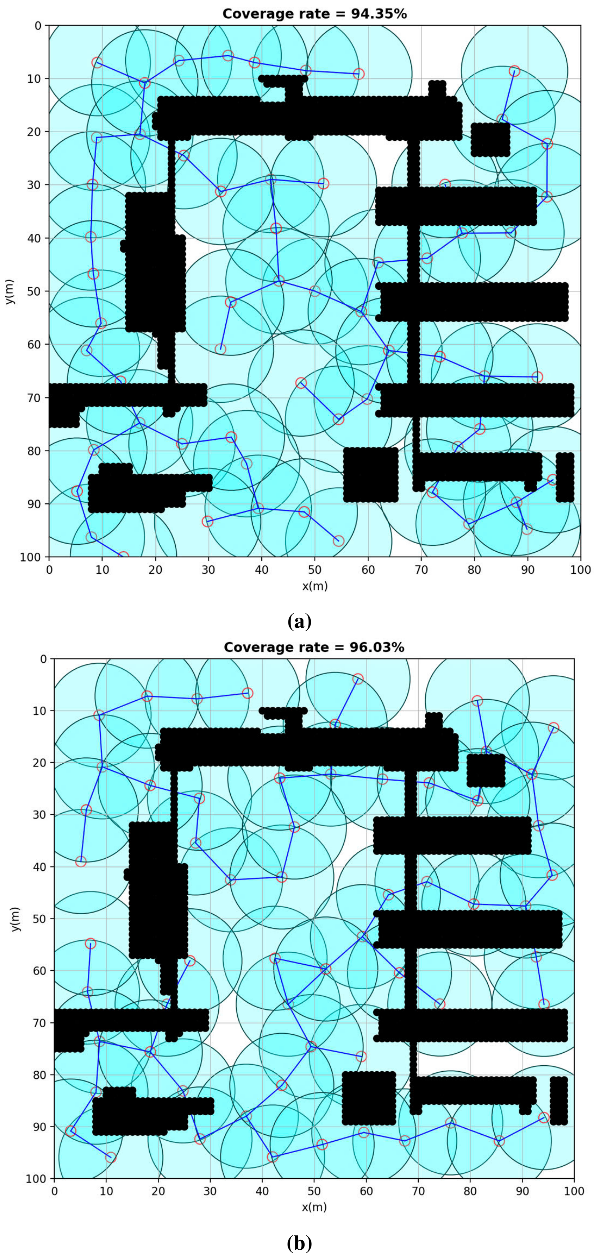


Figure 7: 60 nodes deployment in building obstacles case using (a) FOA (b) NOA

of these regions is larger. On the other hand, the uncovered areas of NOA are insignificant and more scattered in the monitoring region. The NOA algorithm in this case leads the coverage value with 96.03%. Nevertheless, FOA and NOA have both shown their potential for practical applications through diverse types of terrain. In addition, each algorithm presents unique strengths and weaknesses, highlighting the need for a strategic selection depending on the environmental conditions and specific coverage requirements.

4. Conclusion

In this paper, we have successfully applied the FOA and NOA algorithms to solve the CC-CM problem in obstacles-filled environments. Unlike traditional approaches that make use of the communication-sensing radius ratio assumptions, our method accommodates sensors with arbitrary communication ranges and the performance has been validated by comparing with other well-known algorithms. Furthermore, the two algorithms' superior performance are proved through scenarios with both ideal and real-world obstacles, highlighting its practical applicability. However, our simulation of monitoring area is limited to two-dimensional spaces and relies on the binary Boolean disk model, which may not fully reflect sensor characteristics in practice. Future works will focus on extending the proposed approach to three-dimensional spaces and more complex and noisy environments, incorporating sensors with varying radius and more realistic coverage models to enhance the accuracy and applicability of the system.

Acknowledgement

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2024-TĐ-013.

References

[1] H. Baniabdelghany, R. Obermaisser and A. Khalifeh, "Reliable Task Allocation for Time-Triggered IoT-WSN Using Discrete Particle Swarm Optimization," in IEEE Internet of Things Journal, vol. 9, no. 14, pp. 11974-11992, 15 July15, 2022. doi: 10.1109/JIOT.2021.3132452.

[2] K. Zhang, K. Yang, S. Li, D. Jing and H. -B. Chen, "ANN-Based Outlier Detection for Wireless Sensor Networks in Smart Buildings," in IEEE Access, vol. 7, pp. 95987-95997, 2019. doi: 10.1109/ACCESS.2019.2929550.

[3] C. Ma, T. Zuo and H. -F. Zhang, "Network Shortest Path Interdiction Problem Based on Generalized Set Coverage," in IEEE Transactions on Network Science and Engineering, vol. 11, no. 2, pp. 2191-2203, March-April 2024. doi: 10.1109/TNSE.2023.3341452.

[4] 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," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 550-586, Firstquarter 2017. doi: 10.1109/COMST.2016.2610578.

[5] Le. Duc, Oh. Hoon, Yoon. Seokhoon. "Environment Learning-based Coverage Maximization with Connectivity Constraints in Mobile Sensor Networks," IEEE Sensors Journal, 2016. doi: 10.1109/JSEN.2016.2537840.

[6] Z. Hussain, "Metaheuristic Applications and Their Solutions Quality," 2005 International Conference on Information and Communication Technologies, Karachi, Pakistan, 2005, pp. 101-104. doi: 10.1109/ICICT.2005.1598560.

[7] J. Yin, N. Deng, J. Zhang, "Wireless Sensor Network coverage optimization based on Yin–Yang pigeon-inspired optimization algorithm for Internet of Things," Internet of Things, Volume 19, 2022, 100546. doi: 10.1016/j.iot.2022.100546.

[8] Z. Wang, Xie. Huamao, Hu. Zhongdong, Li. Dahai, W. Junling, L. Wen. "Node coverage optimization algorithm for wireless sensor networks based on improved grey wolf optimizer," Journal of Algorithms & Computational Technology, 2019. doi: 10.1177/1748302619889498.

[9] H. ZainEldin, M. Badawy, M. Elhosseini, "An improved dynamic deployment technique based-on genetic algorithm (IDDT-GA) for maximizing coverage in wireless sensor networks," J Ambient Intell Human Comput 11, 4177–4194, 2020. doi: 10.1007/s12652-020-01698-5.

[10] S, Nematzadeh, M. Torkamanian-Afshar, A. Seyyedabbasi, "Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: an efficient metaheuristic-based method for environment-aware node deployment," Neural Comput & Applic 35, 611–641 , 2023. doi: 10.1007/s00521-022-07786-1.

[11] W. Xiaorui, X. Guoliang, Z. Yuanfang, L. Chenyang, P. Robert, G. Christopher, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," SenSys'03: Proceedings of the First International Conference on Embedded Networked Sensor Systems, 2003, 1, 28-39. doi: 10.1145/958491.958496.

[12] M. Abdel-Basset, R. Mohamed, M. Jameel and M. Abouhawwash, "Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems," Knowledge-Based Systems, 262:110248, 2023, pp. 3-12. doi: 10.1016/j.knosys.2022.110248.

[13] Pan. Wen-Tsao, "A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example, Knowledge Based Systems" - KBS, 26, 2012. doi: 10.1016/j.knosys.2011.07.001.

[14] H. N. Minh Vu, S. Tran, D. M. Phan, A. Hoang, and D. C. Hoang, "Maximising Coverage Under Connectivity Constraint Utilising Nature-Inspired Algorithms: A Comparative Analysis," in 2023 Asia Meeting on Environment and Electrical Engineering (EEE-AM), Hanoi, Vietnam: IEEE, Nov. 2023, pp. 01–06. doi: 10.1109/EEE-AM58328.2023.10395009.