Memetic Algorithm for Maximizing K-coverage and K-Connectivity in Wireless Sensor Network

Hanh Nguyen Thi[1], Cuong Van Duc[2], Chinh Tran Duc[2], Hieu Ha Minh[2], Son Nguyen Van[3] and Quan La Van[3]
[1]Faculty of Interdisciplinary Digital Technology, PHENIKAA University, Yen Nghia, Ha Dong, Hanoi, 12116, Vietnam
[2]Hanoi University of Science and Technology, Hanoi, Vietnam
[3]Faculty of Computer Science, PHENIKAA University, Yen Nghia, Ha Dong, Hanoi, 12116, Vietnam
E-mail: hanh.nguyenthi@phenikaa-uni.edu.vn, cuong.vd220021@sis.hust.edu.vn, chinh.td224936@sis.hust.edu.vn, hieu.hm220026@sis.hust.edu.vn, son.nguyenvan@phenikaa-uni.edu.vn, quan.lavan@phenikaa-uni.edu.vn

Keywords: Target coverage, connectivity, node deployment, heuristic algorithm, memetic algorithm, local search, prim algorithm

Received: July 21, 2024

The rapid growth of IoT has enabled diverse applications using Wireless Sensor Networks across various fields. A significant challenge in Wireless Sensor Networks is the efficient deployment of sensors to ensure coverage and connectivity. Effective coverage allows continuous target tracking and data collection, while connectivity ensures data transmission to the base station. In this paper, we address the challenge of maximizing the number of targets satisfying K-coverage and K-connectivity, where each target is tracked by K sensors and has K transmission paths to the base station. We propose a two-phase methodology to tackle this challenge. The first phase enhances the Greedy algorithm to solve the K-coverage problem. The second phase addresses the K-connectivity problem using Memetic algorithms augmented by an efficient local search mechanism called PMA. We evaluate the algorithm on various datasets and compare it with baseline methods, including Greedy and Prim-based with the withdrawal strategy (PWS). Our results show that the proposed PMA with a robust local search outperforms alternative algorithms, with improvements exceeding 10% to 15% compared to the baseline methods. Additionally, we validate the performance of the proposed method using a real-world dataset and outline plans for further enhancements in the near future.

Povzetek: Avtorji so razvili dvofazni pristop za maksimiranje K-pokritosti in K-povezljivosti v brezžičnih senzorskih omrežjih, ki združuje izboljšan pohlepni algoritem in memetični algoritem z lokalnim iskanjem, imenovan PMA.

1 Introduction

Amidst rising environmental concerns, escalating global political tensions, and the widespread proliferation of Internet of Things (IoT) technology and products, particularly emphasizing privacy and security, Wireless Sensor Networks (WSNs) have attracted significant attention[15, 1]. WSNs are composed of sensors equipped with data collection capabilities. Devices must be outfitted with sensor chips capable of detecting environmental phenomena and converting them into accessible data on the Internet for users to analyze and process[19]. These sensors collect data from specific areas and transmit monitoring information to a central base station.

WSNs find crucial applications across various domains, such as military operations, healthcare services, environmental monitoring, biodiversity studies, industrial processes, and urban infrastructure management [20, 7, 12, 5]. For example, wearable, embedded, or ingestible sensors enable continuous monitoring of health parameters and vital signs, such as blood pressure or heart rate, offering vital insights wherever patients or caregivers are situated. The proliferation of WSNs has stimulated significant scientific research and publications aimed at tackling key challenges in sensor networks, including issues related to lifetime, coverage, connectivity, fault tolerance, load balancing, and security. In addition, sensors, characterized by their compact dimensions, face limitations in storage capacity, operational lifespan, and susceptibility to environmental conditions. In areas where battery replacement or recharging is impractical—such as hazardous or obstructed environments like deep oceans or dense forests—ensuring continuous multi-coverage and connectivity between targets and base stations becomes crucial for maintaining network integrity, particularly in scenarios involving potential node failures. Therefore, in this paper, we focus on solving the problem of K-coverage and K-connectivity.

Coverage concern is divided into various subproblems, such as target coverage, area coverage, and barrier coverage. In this study, we focus on resolving the target coverage problem[16, 22]. The target coverage problem involves guaranteeing thorough monitoring of specified targets within a designated surveillance area through strategically placed sensors. The aim is to ensure that every target falls within the sensing range of at least one sensor, enabling comprehensive monitoring and detection capabilities. This challenge is critical across various applications such as environmental monitoring, surveillance, and intrusion detection, where adequate coverage of specific targets is vital for operational efficacy and informed decision-making. The target coverage problem includes 1-coverage, K-coverage, Q- coverage. Within this context, 1-coverage guarantees that all targets are monitored by at least one sensor, K-coverage ensures at least K sensors track each target, and Q-coverage ensures that targets are tracked by Q sensors, the specific value of Q can be adjusted based on priority requirements. In this paper, our primary objective is to resolve the K-coverage problem[8].

Connectivity in WSNs denotes the capacity of sensors to establish and sustain communication links within the network[18]. This ensures reliable data transmission among sensors, facilitating seamless information flow throughout the network. Connectivity is pivotal for fostering collaboration among sensors, streamlining data aggregation, and bolstering network functions like routing and data forwarding. A well-connected network enhances efficiency, resilience, and reliability, enabling effective monitoring and communication across various applications. Connectivity issues encompass 1-connectivity, K-connectivity, and Q-connectivity. In 1-connectivity, a minimum of 1 communication path exists from the target to the base station. K-connectivity guarantees the presence of at least K disjoint paths from the target to the base station[23]. Finally, Q-connectivity ensures the existence of at least Q disjoint paths from the target to the base station, with the value of Q being adjustable based on the target's priority level. In this paper , our primary objective is to resolve the K-connectivity problem.

Recent research endeavors to address multi-coverage and multi-connectivity [10, 3] have encountered limitations, particularly when prioritizing the minimization of sensors required to meet problem constraints, assuming an unlimited number of sensors. Nevertheless, deploying sensors presents substantial hurdles in environments where battery replacement or recharging is unfeasible, such as hazardous or obstructed locations like deep oceans or dense forests. Consequently, the practicality of sensor deployment is constrained, leading to a limited number of sensors in reality. Hence, our team is dedicated to tackling novel problems that, to our knowledge, have yet to be explored by other research groups. Specifically, we focus on determining the maximum number of targets simultaneously fulfilling K-coverage and K-connectivity requirements, given a fixed number of sensors.

In response to the identified challenge, we propose a two-phase strategy. Initially, we aim to resolve the K-coverage issue by refining the Greedy algorithm. Subsequently, the second phase addressed the K-connectivity problem, employing Heuristic and Memetic algorithms augmented with an efficient local search mechanism. The simulation results indicate that the proposed Memetic algorithm combined with Prim and a robust local search function (PMA) outperforms alternative methods, demonstrating superior performance. Therefore, investigating this problem holds scientific and practical significance. In the subsequent section, we present relevant studies concerning this matter.

Our main contributions are listed as follows:

– Formulating a novel problem of K-coverage and K-connectivity suitable for practical application in the 2D domain.

– Presenting a Greedy based method for node deployment that provides K-coverage to all of targets.

– Proposing two baseline methods: PWS and Greedy combined with withdrawal strategy to address connectivity issues.

– Proposing a new approach called PMA (Prim-based Memetic Algorithms): A special Memetic Algorithm Strategy Enhanced with Robust Local Search for Effective Problem Solving.

– Evaluating the proposed method across 40 experimental and real-world datasets.

The rest of the paper is structured as follows: Section 2 provides a comprehensive review of related works. In Section 3, we present the system model and the problem formulation. The proposed algorithms are detailed in Section 4. Section 5 contains the experimental settings, obtained results on various test sets, and a performance comparison with other algorithms to demonstrate the proposal's efficacy. Section 6 discusses conclusions and future.

2 Related work

Coverage and connectivity are two paramount challenges in WSNs. Specifically, coverage in WSNs pertains to the comprehensive monitoring and surveillance of every point within the designated area of interest. [11] The coverage challenge is categorized into three distinct classes based on the intended application: area coverage, target coverage, and barrier coverage [21], [20]. In this paper, our primary focus is on the target coverage predicament, which has been identified as an NP-hard problem. [16] elucidates the various iterations of target coverage.

The emphasis on the NP-completeness of the coverage problem is attributed to the research conducted by [13]. Consequently, most studies advocate solutions employing integer linear programming, heuristic and metaheuristic algorithms to address this challenge. Integer linear programming, which involves constructing a mathematical model, is one of the methodologies employed to resolve the target coverage quandary [3], [23].

However, its effectiveness is evident primarily when dealing with smaller problem sizes, while it demands increased computing time for larger problem sizes [4].

Henceforth, researchers are increasingly delving into the exploration and utilization of heuristic and meta-heuristic algorithms to address the coverage problem. Chien-Chih Liao et al. [14] propose a novel memetic algorithm (MA) that integrates an integer-coded genetic algorithm with local search techniques to solve the K-coverage problem. This approach adapts crossover and mutation operators to integer representation. It introduces a novel fitness function that considers both the number of covers and the individual contribution of sensors to these covers.

When sensors are deployed, a critical consideration arises: determining whether any node in the network can communicate with any other node. Connectivity thus broadens the scope of the coverage problem, aiming to guarantee the existence of pathways between nodes to facilitate the transmission of collected data to external destinations. Moreover, securing network connectivity is paramount for effective WSNs operations. One prevalent approach is to maintain the K-connectivity property, which ensures that the removal of up to K—1 sensor nodes does not lead to network partitioning, thereby preventing the isolation of one or more sensor nodes from the network.

A common tactic to preserve K-connectivity entails adding new nodes as needed. The principal design aim is to reduce the number of additional nodes required while retaining K-connectivity. As with coverage, connectivity poses an NP-hard problem [2] that can be tackled using linear programming and approximation algorithms. One method to address the target connectivity dilemma is integer linear programming, which entails constructing a mathematical model[22]. Nonetheless, its efficacy is more pronounced in managing smaller problem sizes, necessitating escalated computational resources for larger-scale problems. Consequently, researchers are increasingly venturing into exploring and implementing heuristic and meta-heuristic algorithms to tackle the coverage issue. Szczytowski et al. [18] introduced an innovative method for runtime repair and preservation of global WSN K-connectivity, relying solely on localized information. This approach significantly reduces resource demands compared to previous studies.

In recent years, researchers have focused on addressing the challenges of weak security, connection losses during operation, and damaged relay nodes, aiming to ensure dependable monitoring and information transmission. To mitigate these risks, they have specifically targeted solutions for multiple coverage and multiple connections. In [6], Gupta et al. explored a genetic algorithm (GA)-based approach to identify the minimum number of selected potential positions suitable for deploying sensor nodes in target-based wireless sensor networks, ensuring both K-coverage and M-connectivity of the sensor nodes. The study assumes predefined potential positions for sensor node deployment to monitor targets. Similarly, [17] introduces a method based on the Imperialist Competitive Algorithm (ICA), aiming to identify the minimum number of suitable locations for sensor node deployment while meeting coverage and connectivity requirements.

3 System model and problem formulation

3.1 System model

We assume a Wireless Sensor Network and all sensor nodes in it have the same transmission range. Each target collects information from the environment in the range which it is deployed, this region is assumed to be a circular disk whose radius is equal to the sensing range of a sensor node. Then target transmits that information through the sensor nodes on predetermined paths. Transmitting in different paths avoid losing information, if a sensor has problem lead to a path disconnect, there's still other path to transmit information. Two sensor nodes can connect with each other if the Euclidean distance between them is less than or equal the sensing range. Finally, the information is transferred to the Base Station.

3.2 Problem formulation

Let us define surveillance region A as a rectangular with area $W \times H$ and a set T includes m targets $T = \{T_i(x_i, y_i) | 0 \leq x_i \leq W, 0 \leq y_i \leq H, \forall i \in [1, m]\}$. B is the Base Station in A with coordinates (x_B, y_B). We assume set $S = \{S_1, S_2, ..., S_n\}$ is set of n sensors. Our goal is to place n sensors in region A such that maximize the number of targets that satisfied both K-coverage and K-connectivity.

A sensor node can connect with a target if their Euclidean distance is not greater than the sensing range, denoted r_s . Similar, two sensor notes can connect if their Euclidean distance is not greater than the communication range, denoted r_c. Let $c(S_i, T_j)$ denote the connectivity probability between sensor S_i and target T_j, which is calculated via:

$$c(S_i, T_j) = \begin{cases} 1, & \text{if } d(S_i, T_j) \leq r_s, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

and the number of sensors in each target's sensing range is calculated by the following forrmula:

$$C_{T_j} = \sum_{i=1}^{n} c(S_i, T_j). \tag{2}$$

A target T_j is K-coverage if and only if $C_{T_j} \geq K$.

We assume each target, for example T_j, has a set include K path $P = \{P_i | P_i = (T_j, S_{i1}, S_{i2}, ..., S_{il_i}, B), i \in [1, K], l_i$ is number of sensors nodes in $P_i\}$. Then, these K paths will be disjoint if

$$P_a \cap P_b = \{T_j, B\} \; \forall a, b \in [1, K], a \neq b, \quad (3)$$
$$d(T_j, S_{i1}) \leq r_s \; \forall i \in [1, K], \quad (4)$$
$$d(S_{iu}, S_{i(u+1)}) \leq 2r_c \; \forall i \in [1, K], u \in [1, l_i - 1], \quad (5)$$
$$d(S_{il_i}, B) \leq r_c \; \forall i \in [1, K]. \quad (6)$$

Equation (3) ensures that the K paths have no common sensor. Equation (4) make sure that the target and sensors can connect (Their distance are satisfy the sensing range). Equation (5) make sure that the sensors can connect. Equation (6) make sure that the sensors and the Base Station can connect.

A target is K-connectivity if and only if its K paths are disjoint.

Let E_j is the connectivity and coverage status of target T_j. Then

$$E_j = \begin{cases} 1, & \text{if } T_j \text{is both } K\text{-connectivity} \\ & \quad \text{and } K\text{-coverage}, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

From equation (2),(3),(4),(5),(6) and (7) we have problem model:

Maximize

$$\sum_{j=1}^{m} E_j. \quad (8)$$

Subject to

$$C_{T_j} \geq K \; \forall j \in [1, m], \quad (9)$$
$$P_a \cap P_b = \{T_j, B\} \; \forall a, b \in [1, K], a \neq b, \quad (10)$$
$$d(T_j, S_{i1}) \leq r_s \; \forall i \in [1, K], \quad (11)$$
$$d(S_{iu}, S_{i(u+1)}) \leq 2r_c \; \forall i \in [1, K], u \in [1, l_i - 1], \quad (12)$$
$$d(S_{il_i}, B) \leq r_c \; \forall i \in [1, K]. \quad (13)$$

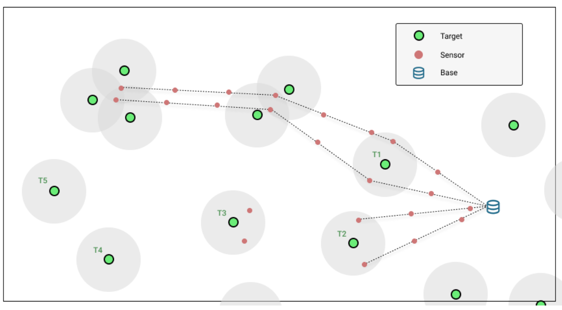

Figure 1: Problem formulation

In **Figure 1**, the targets T_1 and T_2 are characterized by meeting K-coverage and K-connectivity criteria, where K is set to 2. On the other hand, target T_3 achieves K-coverage but fails to meet the K-connectivity requirement. Additionally, targets T_4 and T_5 do not fulfill the K-coverage and K-connectivity criteria.

4 Proposed method

To address the identified challenge, we propose a two-phase methodology. The first phase focuses on resolving the K-coverage issue by enhancing the Greedy algorithm. Subsequently, the second phase is dedicated to tackling the K-connectivity problem, employing Memetic Algorithms augmented by an efficient local search mechanism, PMA. Furthermore, we conduct comprehensive evaluations of the proposed algorithm using various datasets and compare its performance with baseline methods, including Greedy and PWS.

4.1 Coverage phase

In this phase, we aim to determine an optimal approach for placing sensors within region A that can provide K-coverage for each target with the minimum number of sensors. In order to minimize the number of sensors, we apply a Greedy based algorithm.

We consider the set of disks D is the set that conclude the targets which was not satisfied K-coverage. From this set, we construct a set O conclude overlapping regions, which we will use to placed sensors in. An overlapping region is defined by the intersection of the disks in D. And we have a set I that concludes the disks that have no intersection with other disks. For example, in figure below, we have set $O = \{1 \cap 2 \cap 3, 1 \cap 4\}$ and $I = \{5\}$.

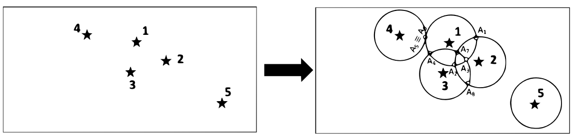

Figure 2: Coverage phase

After having two sets O and I, we will only place sensors in these two sets. At first, we choose a region O_i in O that can cover the most targets. We place K sensors at a random point in it. After placing sensors, if a target satisfies K-coverage, we will remove it from D and update the set O and I. We repeat that procedure until $O = \emptyset$. After placing sensors in O, we start with set I. With every region in I, we repeat placing k sensors at a random point. The entire algorithm is described in **Algorithm 1**.

4.2 Connectivity phase

While targets may meet K-coverage without achieving K-connectivity, and vice versa, these scenarios are not applicable in real-world settings. Therefore, after completing Coverage phase in sensor placement for K-coverage, specialized strategies will optimize sensor positioning. This approach aims to maximize targets by simultaneously meeting both K-coverage and K-connectivity criteria. Based on our understanding, no research studies have addressed the issue we raised. Therefore, we propose the main method, the

Algorithm 1: Greedy based algorithm for coverage

Input : T: Set of targets.
r_s: Sensing range.
n: The number of available sensors
Output : Set R concludes candidate regions.
Location of minimum number of sensor nodes needed to satisfy K-coverage S_2

Build disk set $D = \{D_j(T_j, r_c) | T_j \in T\}$ with each disk has center at one target and radius r_s
$S_2 = \emptyset, R = \emptyset$;
$O \leftarrow$ set of overlapping regions;
$I \leftarrow$ set of targets without overlapping;
while $O \neq \emptyset$ **do**
    $O_i \leftarrow$ Region in O can coverage the most targets;
    $\{s_1, \ldots, s_K\} \leftarrow$Place K sensors at a random point at O_i;
    $R = R \cup O_i, S_2 = S_2 \cup \{s_1, \ldots, s_K\}$;
    if *a target T_j satisfies K-coverage* **then**
        Remove D_j from D;
        Update O, I;
    end
end
for *region* $\in I$ **do**
    $\{s_1, \ldots, s_K\} \leftarrow$Place K sensors at a random point in region ;
    $R = R \cup O_i, S_2 = S_2 \cup \{s_1, \ldots, s_K\}$;
end
return S_2, R;

Prim-based Memetic Algorithms (**PMA**) in Section 4.2.3, alongside two baseline methods, **Greedy** and Prim-based with withdrawal strategy **PWS** in Section 4.2.1, 4.2.2, for comparative analysis.

4.2.1 Greedy based algorithm for connectivity (Greedy)

At first, we introduce a Greedy based algorithm used for maximize the number of targets satisfy both K-coverage and K-connectivity by a limited number of sensors. After coverage phase, we assume number of available sensors is $n_a = n - |S|$. However, in this phase, when we exhaust all of the sensors, we will remove sensors from special regions within coverage phase to optimize the result.

We assume set U concludes the base station and the sensors which is available to connect to the base station and set L concludes the regions which unsatisfactory K-connectivity. Beginning, we initialize $U = \{B\}$ and $L = R$. Each region in L, U is represented by the location of the sensors in it. We define the region in L which has the shortest distance to a point in U. Then, we place sensors in order to connect that region with U. To connect two regions, from a point in first region, we create a path to the nearest point which not have path in the second region and repeat until we have K separate paths between two regions. The sensors amount need to connect two points I and J (I, J can be Base Station or sensor) is calculates by

$$sensor\_amount = [\frac{d(I,J) - 2r_c}{2r_c}] + 1. \quad (14)$$

where [.] denotes the integer part of a number.

If in this process, we exhaust all of the sensors, we will choose the region that have furthest distance to a point in U and remove all sensors in it to use for connect until the sensors amount is enough to connect or there are no more sensors to remove. If the available sensor can not increase the number of target satisfy K-coverage and K-connectivity, we will place them in some region that has density of sensors highest aim to enhance connectivity.

The entire algorithm is described in **Algorithm 2**.

Algorithm 2: Greedy based algorithm for connectivity

Input : R: Set of candidate regions.
S_2: Set of sensors use for coverage.
n_a: The number of available sensors
Output : The number of targets satisfy both K-coverage and K-connectivity

$U = \{B\}$ Connected set;
$L = R \leftarrow$ Unconnected set;
$res \leftarrow$ Number of target satisfy K-coverage and K-connectivity;
while $n_a > 0$ *or* $|L| > 0$ **do**
    $min_{dist} = +\infty$;
    for $i \in L$ **do**
        for $j \in U$ **do**
            if $d(i, j) < min_{dist}$ **then**
                $min_{dist} = d(i, j)$;
                $l_{choose} = i, u_{choose} = j$;
            end
        end
    end
    $sensor_{need} \leftarrow$ Sensors amount need to connect l_{choose}, u_{choose}
    while $n_a < sensor_{need}$ *and* $|L| > 1$ **do**
        Choose region in V furthest to U and remove from L;
        $n_a = n_a + K$;
    end
    if $n_a \geq sensor_{need}$ **then**
        Place sensors;
        Update L, U, res, n_a;
    end
    else
        Remove sensors in L; $n_a = n_a + K$;
        Place n_a sensors in high sensor density region;
    end
end
return res;

4.2.2 Prim-based with withdrawal strategy (PWS)

First, each region is represented by the location of the sensors in it. We consider a graph $G = (V, E)$ where V conclude Base Station and candidate regions $V = B \cup R$ and $E = \{(u, v, dist(u, v)|u, v \in V, u \neq v\}$ with $dist(u, v)$ is the number of sensors required to connect u to v by K node-disjoint paths calculate by the formula has been presented before. We will apply Prim's algorithm to find the Minimum Spanning Tree starts from B. We define set $S_l = \{v|v \text{is leaf node}\}$. n_{left} is the number of sensors left after placing sensors to the tree (n_{left} can be negative if sensor amount is not enough) calculate by $n_{left} = n_a - n_{MST}$ with n_{MST} is the number of sensor need to placing in Minimum Spanning Tree. Until $n_{left} \geq 0$, we choose node v in S_l that have $dist(v, parent(v))$ is the biggest, remove v from s_l and update n_{left}, s_l. The number of satisfied target is the number of remaining node in the tree (except B).

The entire algorithm is described in **Algorithm 3**.

Algorithm 3: Prim-based with withdrawal strategy(PWS)

Input : R: Set of candidate regions.
S: Set of sensors use for coverage.
n_a: The number of available sensors

Output : The number of targets satisfy both K-coverage and K-connectivity

```
Graph G = (V, E) with V = B ∪ R , E = {(u, v, dist(u, v)|u, v ∈ V, u ≠ v};
Apply Prim's algorithm to find the Minimum Spanning Tree;
S_l ← Set of leaf node;
n_left ← the number of sensors left after placing sensors to the tree;
while n_left < 0 do
    v ← Leaf node has biggest dist(v, parent(v));
    Remove v from S_l;
    update S_l, n_left
end
res ← The number of remaining node (except B);
return res
```

4.2.3 Prim-based memetic algorithms (PMA)

Our proposed method presents an innovative and efficient approach to solving the problem. Each solution element is encoded as a binary vector, representing whether a specific area is utilized for connectivity. This encoding serves as the basis for generating the initial population. We introduce two advanced strategies for crossover and mutation operators to enhance the evolutionary process. These strategies are designed to direct new individuals toward promising regions in the solution space while preserving population diversity, thereby expediting convergence to optimal solutions. Furthermore, a local search mechanism is integrated to refine the best-performing individuals, increasing the potential to escape local optima. A distinctive evaluation mechanism is employed in which unsatisfying individuals are not immediately discarded. Instead, they are retained and evaluated using a specialized strategy, ensuring consistent population diversity throughout the optimization process.

Detailed explanations of each component in the proposed method are provided in the following sections.

4.2.3.1 Individual representation A individual is a vector of integers of size $n + 1$, where n denotes the number of coverage areas. We incremented the count by 1 to incorporate the Base station. Our research paper defines a individual as a significant binary sequence comprising 0s and 1s. Here, a 1 denotes the location of the associated coverage area for establishing the connecting line. Conversely, a 0 indicates the corresponding coverage area where the response is negated, thus disregarding any potential connection path. For example if $n = 5$, a individual can can be $c_1 = [110100]$, this represent that areas $1, 3$ are considered to find the connection to Base station.

4.2.3.2 Genetic operators In this paper, we employ a novel crossover and mutation heuristic strategy along with a potent local search function to seek optimal results.

Crossover : With two random chromosomes from the population, we denoted them as P_1, P_2 . Then we introduce the following heuristic crossover method for generating new chromosomes C from P_1 and P_2.

$$C[i] = \begin{cases} P_1[i], & \text{if } P_1[i] = P_2[i], \\ P_1[i], & \text{if } P_1[i] \neq P_2[i] \\ & \quad \text{and } p < \frac{fitness(P_1)}{fitness(P_1)+fitness(P_2)}, \\ P_2[i], & \text{otherwise} \end{cases} \tag{15}$$

where p is a random number in range $[0, 1]$.

Mutation : We propose a Heuristic Mutation. A chromosome satisfies when it has enough sensors for connectivity. For these chromosomes, we will iterate through points with a value of 0 and change them to 1 with a given probability α (based on experimentation). Each time there's a change, decrease α by an amount of $\frac{1}{2n}$ to ensure there are not too many changes (as excessive changes can lead to violations). Similarly for chromosomes that do not satisfy, we apply the same strategy but instead of changing 0 to 1, we change 1 to 0.

Local search : With the chromosome that has the best fitness P_{best}, we will iterate through all points in P_{best} and replace each value of 0 with 1. If a new chromosome that satisfies the conditions is generated, this will be the new best chromosome.

4.2.3.3 Evaluation The fitness value of a chromosome A is determined according to a special strategy as follows:

$$fitness(A) = \begin{cases} |T|, & \text{if enough sensors for connectivity} \\ \frac{1}{|U|}, & \text{otherwise} \end{cases} \quad (16)$$

with $|T|$ is the number of targets that satisfy K-coverage and K-connectivity and $|U|$ is the number of missing sensors.

In our study, we retain unsatisfactory chromosomes within the population to preserve potentially beneficial genetic material for subsequent generations. Notably, individuals with fewer sensor deficiencies are assigned higher fitness values than those with greater deficiencies. To consistently meet the constraint, individuals of higher quality are assigned elevated fitness values (If chromosome X outperforms chromosome Y, then $fitness(X) > fitness(Y)$).

4.2.3.4 Selection and replacement Starting from a population denoted as P comprising N elements, we will generate a new population, labeled P_{new} also consisting of N chromosomes, employing a specialized heuristic strategy :

Initially, the individuals within P will be arranged in descending order based on their fitness value.

We defined two probabilities p_1, p_2. Where p_1 decides whether we will use crossover or mutation and p_2 to decide whether we will do with the whole population or with some top chromosomes. New chromosomes are created and added to P_{new} until $|P_{new}| = N$.

Next, we merged populations P and P_{new} to form P_{mix}, then sorted P_{mix} in descending order of fitness values. Subsequently, to construct the potential population P_p, we select individuals as follows: Initially, the top c_{top}% elements of P_{mix}, representing the best individuals, are chosen and removed. Next, the remaining aspects of P_{mix} are shuffled, and $c_{routlette}$% elements are selected using the roulette wheel selection method. And then $P = P_p$. After that, choosing randomly c_{loc}% elements in P to undergo local search. This process iterated max_gen times.

Function $bestfitness(P)$ return the individual with the highest fitness value in P.

The entire algorithm is described in **Algorithm 4**.

Algorithm 4: Prim-based Memetic Algorithms (PMA)

Input : R : Set of candidate regions.
n_a : The number of available sensors.
N : The number of individuals in a population.
max_gen : The number of generation.
p_1, p_2 : Mutation coefficient
Output : The number of targets satisfy both K-coverage and K-connectivity

$P \leftarrow$ Randomly generate N individuals;
$count = 0$
while $count < max\_gen$ **do**
  $P_{new} = \emptyset$
  while $|P_{new}| < N$ **do**
    if $p_1 < m$ **then**
      $x = crossover(P_1, P_2)$ if $p_2 < m_1$
      $y = mutation(P_1)$ if $p_2 < m_2$
      where P_1, P_2 is randomly in P
    end
    else
      $x = crossover(P_1, P_2)$ if $p_2 < m_1$
      $y = mutation(P_1)$ if $p_2 < m_2$
      where P_1, P_2 are randomly selected from the top N_{best} elements in P.
    end
    $P_{new} \cup x \cup y$
    p_1, p_2 is random numbers in range $[0, 1]$
  end
  $P_{mix} = P + P_{new}$
  Calculate fitness value of every individual in P_{mix}.
  X = Select c_{top}% of elements in P_{mix} with the highest fitness values.
  $P_{mix} \setminus X$
  X_1 = Choose $c_{routlette}$% elements using the roulette wheel selection method in P_{mix}
  $P = X + X_1$, Apply local search for randomly c_{loc}% elements in P
  $count = count + 1$
end
x = local search(x) , $x$in P
Return $bestfitness(P)$;

5 Numerical results

5.1 Parameter setting

Our algorithms are implemented in Python and executed on Visual Studio Code with Intel(R) i5-12500H 3.1GHz CPU, RAM 16GB DDR4 1600MHz.

The parameter is configured for presentation inTable 1.

Table 1: Parameter value for PMA

Parameter	Value
Population size (N)	200
N_{best}	50
Number of generations (max_gen)	300
Crossover rate($p_1 = p_2$)	20%
c_{top}	25
$c_{routlette}$	75
c_{loc}	10
α	0.3

5.2 Problem instances

Due to the lack of public research related to this problem, we conducted an experiment on a new dataset consisting of four scenarios for both phases: K-coverage and K-connectivity fromTable 1 . The data set is limited to the 1000×1000 domain. We randomly generate the locations of targets and Base stations in surveillance region A of size $1000 \times 1000(m^2)$ with uniform distribution.

We have 4 scenarios:

scenario 1 The scenario includes 10 instances as given inTable 2; each instance undergoes execution across 10 distinct test sets, followed by averaging, to assess the impact of the number of sensors on solution quality.

scenario 2 The scenario includes 10 instances as given inTable 3; each instance undergoes execution across 10 distinct test sets, followed by averaging, to assess the impact of the number of targets on solution quality.

scenario 3 The scenario includes 5 instances as given inTable 4; each instance undergoes execution across 10 distinct test sets, followed by averaging, to assess the impact of the number of K on solution quality.

scenario 4 The scenario includes 10 instances as given inTable 5; each instance undergoes execution across 10 distinct test sets, followed by averaging, to assess the impact of the number of r on solution quality.

Table 2: Parameter values for test instances in scenario 1

Dataset	n	m	r	K	$A(W \times H)(m^2)$
s1-1	400				
s1-2	440				
s1-3	480				
s1-4	520				
s1-5	560	150	20	3	1000×1000
s1-6	600				
s1-7	640				
s1-8	680				
s1-9	720				
s1-10	760				

Table 3: Parameter values for test instances in scenario 2

Dataset	n	m	r	K	$A(W \times H)(m^2)$
s1-1		60			
s2-2		70			
s2-3		80			
s2-4		90			
s2-5	400	100	20	3	1000×1000
s2-6		110			
s2-7		120			
s2-8		130			
s2-9		140			
s2-10		150			

Table 4: Parameter values for test instances in scenario 3.

Dataset	n	m	r	K	$A(W \times H)(m^2)$
s3-1				1	
s3-2				2	
s3-3	400	150	20	3	1000×1000
s3-4				4	
s3-5				5	

Table 5: Parameter values for test instances in scenario 4

Dataset	n	m	r	K	$A(W \times H)(m^2)$
s4-1			12		
s4-2			14		
s4-3			16		
s4-4			18		
s4-5	400	150	20	3	1000×1000
s4-6			22		
s4-7			24		
s4-8			26		
s4-9			28		
s4-10			30		

5.3 Experiment results

We run 4 scenarios on the dataset and evaluate the obtained results. For this evaluation, we used the variable score $= \frac{E}{n}$, where E represents the number of targets satisfying K coverage and K connectivity, and n denotes the total number of targets. The detailed results are presented below.

In this experiment, domain A is a large region of size 1000 × 1000. Result of this experiment is given in Fig 3, Fig 4, Fig 5 and Fig 6. With the dataset we have, it is clear that there is a significant distance between nodes, which means there are not many overlapping regions. In comparing two base methods, their results exhibit a notable similarity, whereas the proposed method PMA consistently demonstrates superior performance over both base methods.

scenario 1

In this experimental setup, the value of n was incrementally raised from 400 to 760 to investigate the influence of the sensor count on the outcomes generated by three distinct algorithms. The result is shown in Figure 3. The analysis reveals that PMA surpasses the performance of the two base methods. To be precise, PMA exhibits a superiority of 110% over PWS and Greedy algorithms. In expansive spatial contexts, PMA demonstrate enhanced efficacy relative to baseline methods, owing to the integration of heuristic crossover, mutation mechanisms, and extensive local search. As the number of sensors increases, the opportunity for targets to meet both K-coverage and K-connectivity requirements rises significantly

scenario 2

In our experimental setup, we incrementally varied the value of m from 60 to 150 to study how the number of

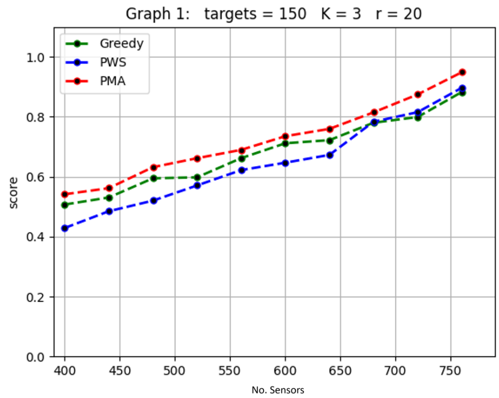

Figure 3: Impact of the number of sensor on PMA, PWS and Greedy

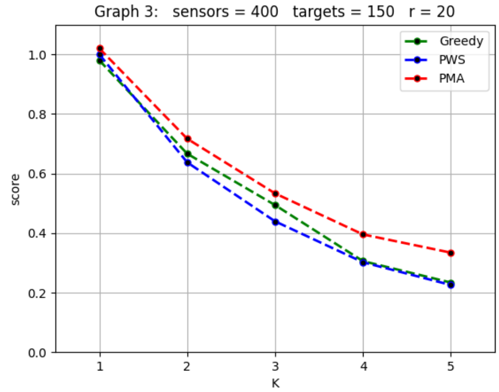

Figure 5: Impact of the number of sensor on PMA, PWS and Greedy

targets influences the outcomes produced by three different algorithms. Figure 4 illustrates the results, showing that PMA outperforms both baseline methods. Specifically, PMA demonstrates a superiority of 113% over the PWS and Greedy algorithms in terms of performance. As the number of targets increases, more sensors are needed to cover and connect all targets in the region. Consequently, the number of targets satisfying the constraints decreases.

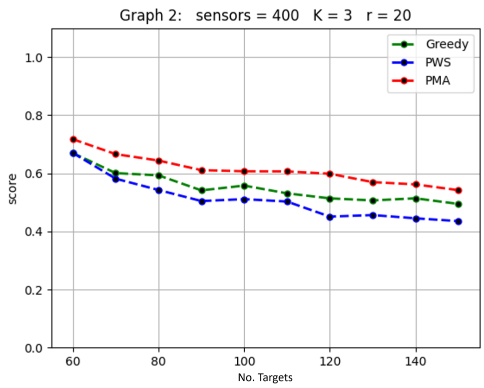

Figure 4: Impact of the number of sensor on PMA, PWS and Greedy

scenario 3

In this experiment, we varied K from 1 to 5 to assess its influence on the outcomes produced by three algorithms. The results depicted in Figure 5 demonstrate that PMA outperforms both baseline methods, showing a performance advantage of 115% over the PWS and Greedy algorithms. To explain why PMA outperforms, as the value of K increases, devising an effective sensor deployment strategy for optimal solutions becomes more challenging. However, PMA maintains its strength through diverse exploration of feasible solution spaces, consistently approaching nearly optimal outcomes.

scenario 4

In this experiment, r ranged from 12 to 30 to examine its impact on the outcomes of three algorithms. The findings in Figure 6 indicate that PMA surpasses the performance of the two baseline methods, exhibiting a superiority of 124% over Prim and Greedy algorithms. As r increases, sensor deployment creates more overlapping areas, which reduces the required number of sensors and thereby increases the number of targets meeting both K-coverage and K-connectivity criteria. Furthermore, more significant overlap introduces various deployment strategies, and through its diverse exploration of solution space, PMA demonstrates superior performance compared to baseline methods.

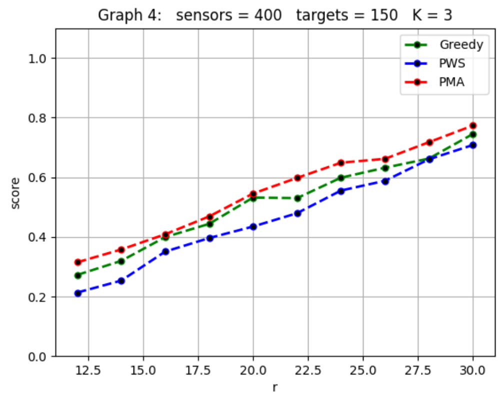

Figure 6: Impact of the number of sensor on PMA, PWS and Greedy

5.4 Real-world dataset

In this section, we utilize a real-world dataset to assess the performance of the PMA method. The original coordinate data were sourced from [9]. This dataset comprises the coordinates of 43 targets, representing railway stations and bus stops near the center of Hanoi, the capital of Vietnam. The base station is positioned at the Vietnam Academy of Agriculture. To facilitate analysis, we normalized the coordinates to a range of $[0, 1000]$.

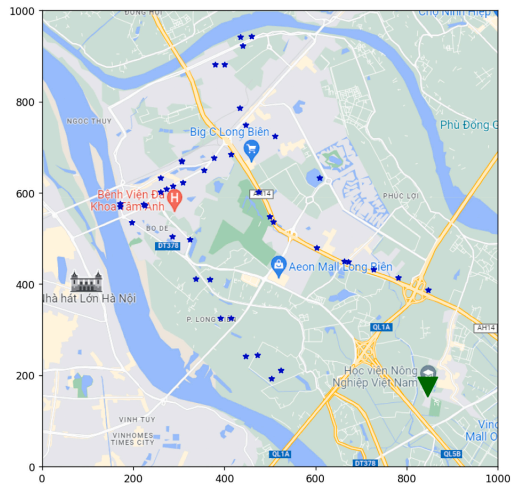

Figure 7: Target locations on the map

The illustration for the target coordinates is depicted in Figure 7. Suppose we aim to monitor these targets from the academy utilizing a WSN. We consider the following two scenarios:

- scenario 1: Let $K = 2$, which means the targets require low resources.
- scenario 2: Let $K = 5$, which means the targets require high resources.

All other parameters remain constant: $r_c = r_s = 60m$.

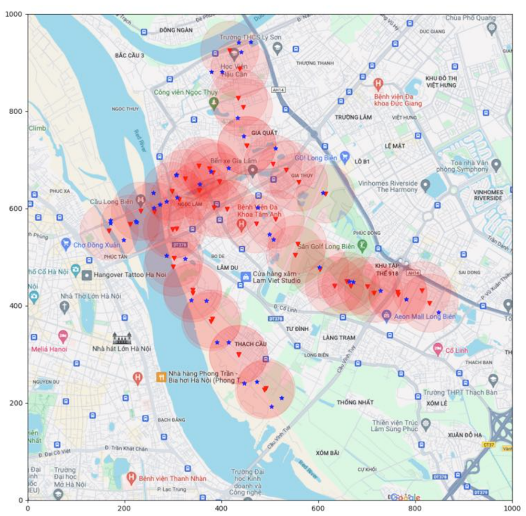

Figure 8: Result of scenario 1

The outcomes of the two scenarios are depicted in Figures 8, and 9. In these figures, blue stars denote targets,

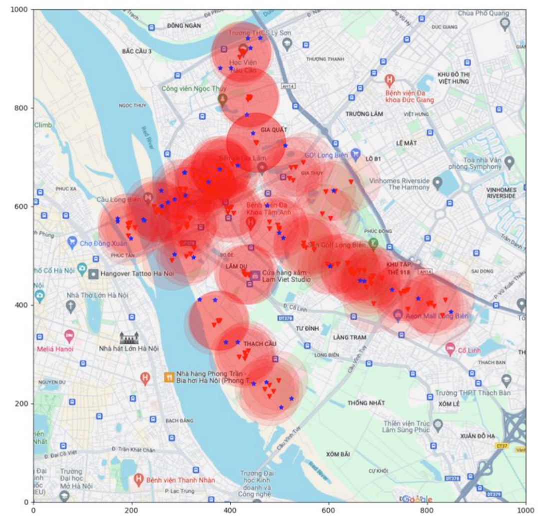

Figure 9: Result of scenario 2

while red triangles indicate sensor nodes. In scenario 1, the PMA algorithm requires a total of 50 sensors and operates in 0.033 seconds. Conversely, scenario 2 requires 125 sensors and runs in 0.054 seconds. It is evident that selecting optimal regions in Phase 1 allows PMA to deploy sensors effectively within the intersection areas of the target disks. Furthermore, optimizing connections in Phase 2 helps reduce the consumption of sensor nodes.

6 Conclusion

This paper presents a model that maximizes the number of targets satisfying K-coverage and K-connectivity with a fixed number of sensors. The problem is addressed in two phases: the first phase optimally places sensors to achieve K-coverage, while the second phase establishes optimal connections to ensure K-connectivity. The Greedy algorithm is proposed to solve the first phase, while a novel method called PMA is employed for the second phase and compared with Prim and Greedy algorithms. Extensive testing across four scenarios reveals that optimizing K-coverage and K-connectivity significantly impact network deployment. The proposed PMA outperforms existing Prim and Greedy methods.

These findings promise future advancements in Wireless Sensor Networks. In the future, we plan to further study this problem and consider more factors such as obstacles, energy efficiency and network lifetime, clustering and routing, deployment in 3D environment

Acknowledgement

This research is funded by Ministry of Education and Training under project number B2024.NHF.01.

References

[1] Nguyen Thi My Binh et al. "An efficient exact method with polynomial time-complexity to achieve k-strong barrier coverage in heterogeneous wireless multimedia sensor networks". In: *Journal of Network and Computer Applications* 231 (2024), p. 103985. URL: https://doi.org/10.1016/j.jnca.2024.103985.

[2] Jonathan L. Bredin et al. "Deploying sensor networks with guaranteed capacity and fault tolerance". In: (2005). DOI: https://doi.org/10.1145/1062689.1062729.

[3] Dina S. Deif and Yasser Gadallah. "Classification of wireless sensor networks deployment tech niques". In: (2013). DOI: https://doi.org/10.1109/SURV.2013.091213.00018.

[4] S.S. Dhillon, K. Chakrabarty, and S.S. Iyengar. "Sensor placement for grid coverage under imprecise detections". In: (2002). DOI: https://doi.org/10.1109/ICIF.2002.1021005.

[5] Liang Du. "Design and Application of Intelligent Building Environment Monitoring System Based on Wireless Sensor Network". In: *Informatica* 48.15 (2024). URL: https://doi.org/10.31449/inf.v48i15.6514.

[6] Suneet Kumar Gupta, Pratyay Kuila, and Prasanta K. Jana. "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks". In: (2016). DOI: https://doi.org/10.1016/j.compeleceng.2015.11.009.

[7] N. T. Hanh et al. "Minimal Node Placement for Ensuring Target Coverage With Network Connectivity and Fault Tolerance Constraints in Wireless Sensor Networks". In: (2019). DOI: http://dx.doi.org/10.1109/CEC.2019.8789961.

[8] Nguyen Thi Hanh et al. "Minimal Relay Node Placement for Ensuring Network Connectivity in Mobile Wireless Sensor Networks". In: (2021). DOI: https://doi.org/10.1109/NCA51143.2020.9306727.

[9] Nguyen Thi Hanh et al. "Node placement optimization under Q-Coverage and Q-Connectivity constraints in wireless sensor networks". In: (2023). DOI: https://doi.org/10.1016/j.jnca.2022.103578.

[10] Nguyen Thi Hanh et al. "Optimizing wireless sensor network lifetime through K-coverage maximization and memetic search". In: (2023). DOI: https://doi.org/10.1016/j.suscom.2023.100905.

[11] Banu Kabakulak. "Sensor and sink placement, scheduling and routing algorithms for connected coverage of wireless sensor networks. Ad Hoc Networks". In: (2019). DOI: https://doi.org/10.1016/j.adhoc.2018.11.005.

[12] Dionisis Kandris et al. "Applications of wireless sensor networks: An up-to-date survey". In: (2020). DOI: https://doi.org/10.3390/asi3010014.

[13] Wei-Chieh Ke, Bing-Hong Liu, and Ming-Jer Tsai. "The critical-square-grid coverage problem in wireless sensor networks is NP-complete". In: (2011). DOI: https://doi.org/10.1016/j.comnet.2011.03.004.

[14] Chien-Chih Liao and Chuan-Kang Ting. "A Novel Integer-Coded Memetic Algorithm for the Set k-Cover Problem in Wireless Sensor Networks". In: (2017). DOI: https://doi.org/10.1109/TCYB.2017.2731598.

[15] S. Pundir adn M. Wazid et al. "Intrusion detection protocols in wireless sensor networks integrated to internet of things deployment: Survey and future challenges". In: (2020). DOI: http://dx.doi.org/10.1109/ACCESS.2019.2962829.

[16] S. Mini, Siba K. Udgata, and Samrat L. Sabat. "Sensor deployment and scheduling for target cover age problem in wireless sensor networks". In: (2013). DOI: https://doi.org/10.1109/JSEN.2013.2286332.

[17] Hemmat Sheikhi and Wafa Barkhoda. "Solving the k-Coverage and m-Connected Problem in Wireless Sensor Networks through the Imperialist Competitive Algorithm". In: (2020). DOI: https://doi.org/10.1142/S0219265920500024.

[18] Piotr Szczytowski, Abdelmajid Khelil, and Neeraj Suri. "DKM: Distributed k-Connectivity Maintenance in Wireless Sensor Networks". In: (2012). DOI: https://doi.org/10.1109/WONS.2012.6152244.

[19] A. Tripathi et al. "Coverage and Connectivity in WSNs: A Survey, Research Issues and Challenges". In: (2018). DOI: https://doi.org/10.1109/ACCESS.2018.2833632.

[20] Bang Wang. "Coverage control in sensor networks". In: (2010). DOI: http://dx.doi.org/10.1007/978-1-84996-059-5.

[21] Bang Wang. "Coverage problems in sensor networks: A survey". In: (2011). DOI: https://doi.org/10.1145/1978802.1978811.

[22] Huping Xu, Jiajun Zhu, and Bang Wang. "On the deployment of a connected sensor network for confident information coverage". In: (2015). DOI: https://doi.org/10.3390/s150511277.

[23] Xiaochun Xu and Sartaj Sahni. “Approximation algorithms for sensor deployment”. In: (2008). DOI: https://doi.org/10.1109/TC.2007.1063.