HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

GRADUATION THESIS

SPARTA-GEMSTONE: A two-phase approach for efficient node placement in 3D wireless sensor networks under Q-Coverage and Q-Connectivity constraints

VŨ QUANG TRƯỜNG
truong.vq194198@sis.hust.edu.vn

Major: Information Technology
Specialization: Computer Science

Supervisor: Ph.D Trịnh Văn Chiến _____

Signature

Department: Computer Science

School: School of Information and Communications Technology

HANOI, 06/2023

ACKNOWLEDGMENT

First and foremost, I would like to express my heartfelt gratitude to MSO Lab and CV Lab, where my passion for research was nurtured and developed. I am deeply indebted to **Associate Professor Huynh Thi Thanh Binh**, **PhD Trinh Van Chien**, **PhD Nguyen Thi Hanh**, **Nguyen Van Son**, **Tran Thi Uyen**, and **M.Sc. Le Van Quan** of MSO Lab and **PhD Nguyen Thi Oanh** of CV Lab for their unwavering guidance and support throughout my research journey. Additionally, I extend my sincere thanks to **Trinh The Minh** and **Do Minh Tam** for their invaluable assistance in my thesis. Their contributions were indispensable in the completion of this work.

Furthermore, I am profoundly grateful to my beloved family, including my parents, my brother, and all other family members. Their constant presence and great support have been crucial in every decision I have made, making them my most significant source of mental strength. Without their love and encouragement, none of this would have been possible.

I would also like to acknowledge my dear friends and allies at HUST, particularly **Nguyen Phuc Tan**, **Nguyen Ngoc Bao**, **Nguyen Trong Bang**, **Nguyen Quoc Hao**, and **Nguyen Trung Hieu**. Their trust and support throughout my university journey have been invaluable. I am truly grateful for their friendship and selfless assistance, given without any hesitation or expectation of anything in return.

My four-year journey at HUST has been nothing short of fantastic and remarkable, and it is all thanks to each and every one of them. Finally, I want to express my gratitude to everyone who has been a part of my journey. Whether we are still connected or have gone our separate ways, and irrespective of the joys or sorrows we have shared, each experience has played a crucial role in shaping me into a better version of myself, creating an indelible journey that I will always cherish.

Cảm ơn Bách Khoa!

Truong Vu

ABSTRACT

Target coverage and connectivity problems are major challenges for wireless sensor networks (WSNs). In practice, when different targets demand different priority levels, each target is assigned a value q which is the number of sensors covering it as well as the number of node-disjoint paths to transfer sensing data of itself to the base station. When $q > 1$, the network can ensure fault tolerance. Those constraints are named Q-Coverage and Q-Connectivity. Because of the practicality of the problem, various pieces of research are attempted to tackle it. Nevertheless, existing works do not directly aim to minimize the number of nodes under these both constraints simultaneously in realistic 3D environment. Moreover, the type of connection mentioned in most of these works is the connection between sensors, which is not suitable for target-oriented networks.

Since this NP-hard problem involves two constraints, this thesis proposes a two-phase solution. In Phase I, we utilize a heuristic algorithm called **S**phere **P**oint **A**nd **R**emoval **T**arget **A**lgorithm (SPARTA) with two variances SPARTA-CC and SPARTA-CP to address one of the constraints. In Phase II, we employ a heuristic algorithm based on Minimum Spanning Tree (MST) called **G**raph-based **E**fficient **M**inimum **S**panning **T**ree **O**ptimization for **N**etwork **E**stablishment (GEMSTONE) to tackle the other constraint. Additionally, we evaluate the performance of these algorithms using a proposed 3D dataset and compare them with baseline methods, namely GLA in Phase I, CMFA and FCSA in Phase II.

The thesis introduces a new problem in 3D WSNs related to target coverage & connectivity and presents a novel method along with a dataset in two distinct levels of height variation to address it. The results indicate a significant improvement in various evaluation metrics when compared to the baseline methods. These findings are not only beneficial for future research on WSNs but also specifically contribute to the advancement of target coverage.

Overall, this thesis provides valuable insights and practical contributions to the field of WSNs, highlighting improved algorithms and their effectiveness in enhancing target coverage.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
BDCM	Binary Disk Coverage Model
GEMSTONE	Graph-based Efficient Minimum Spanning Tree Optimization for Network Establishment
MST	Minimum Spanning Tree
SPARTA	Sphere Point And Removal Target Algorithm
WSN	Wireless Sensor Network

CHAPTER 1. INTRODUCTION

The issues of coverage and connectivity have garnered considerable attention in academic research due to their extensive range of applications. This chapter provides an overview of the fundamental context of the problem addressed in the thesis, along with outlining its contributions and organizational structure.

1.1 Background

In recent years, the Internet of Things (IoT) has garnered significant attention from the general public and the technology community worldwide. One of the essential components of IoT is Wireless Sensor Networks (WSN). Devices need to be equipped with sensor chips to detect phenomena in the physical environment and convert them into data on the Internet for user processing and analysis [1], [2]. Therefore, wireless sensor networks contribute significantly to the success of IoT [3]. Sensors in the network gather data from a specific area and transmit monitoring data to a base station. Subsequently, processing units analyze the data and generate reports to provide meaningful information to users.

WSN finds applications in various fields, including military, healthcare, environmental monitoring, agriculture, industry, transportation, and more [4], [5]. For instance, sensors can be embedded in the human body, on the skin, or on clothing items to monitor health conditions and measure vital signs such as blood pressure, heart rate, blood oxygen levels, body temperature, and glucose levels, enabling appropriate and timely treatment for patients. Additionally, WSN is used to monitor environmental conditions in healthcare facilities, such as temperature, humidity, light, dust, toxic gases, noise, etc. It helps maintain a safe and comfortable working environment and ensures optimal patient care. With the value that WSN brings, there is an increasing number of research and scientific works published to address challenges in sensor networks, such as network lifetime, coverage capacity, connectivity, fault tolerance, load balancing, and security.

Coverage issues can be categorized into three types: target coverage, area coverage, and barrier coverage. This thesis focuses on addressing the target coverage problem (ensuring all targets are covered by at least one sensor) using the Binary Disk Coverage Model (BDCM) [6]. BDCM states that a sensor s with sensing range r_s covers an target t if and only if the Euclidean distance between t and s is less than r_s. The target coverage problem is divided into three subproblems: 1-coverage (simple coverage), k-coverage, and Q-coverage. In 1-coverage, each target must be covered by at least one sensor. However, in practice, sensors are prone to

errors and failures. Hence, the k-coverage model is proposed to ensure fault tolerance in the network, where each target requires at least k sensors for coverage [7], [8]. Nevertheless, certain applications require different targets to have varying degrees of coverage priority, specifically, the minimum number of nodes needed to monitor each target may differ. This is referred to as Q-coverage [9], [10] , which is the focus of this research problem.

Network connectivity issues also have constraints similar to coverage, specifically, 1-connectivity, k-connectivity [11]–[14], and Q-connectivity. Connections are established to transmit data from targets to the base station through relay nodes. Each relay node has a certain communication range. The Q-connectivity constraint states that each target must have q non-intersecting paths to the base station, and the number of such paths may vary for different targets, similar to Q-coverage.

To the best of the our knowledge, the Q-coverage problem in two-dimensional space has been addressed in various sensor network types such as wireless sensor networks, directional sensor networks [15]–[20]. However, authors consider that addressing this problem in a 2D deployment domain is not suitable for several practical applications such as monitoring targets in smart cities, IoT systems, where smart devices equipped with sensors need to be monitored in real-time and placed in a 3D space.

Regarding coverage and connectivity issues in a 3D space, several studies have tackled this problem [21], [22]. They addressed the Q-coverage by vertex coloring algorithm and 1-connectivity problems by Breadth-first search.

Based on the existing research, we realize that studying the problem of minimizing the number of deployed nodes to satisfy Q-coverage and Q-connectivity constraints simultaneously in a 3D environment is highly suitable for numerous optimized IoT system deployments. Currently, no research has tackled this specific problem. Therefore, this thesis focuses on addressing the problem of finding the positions of sensor nodes to satisfy Q-coverage and Q-connectivity constraints with a minimal number of deployed nodes in a 3D terrain, aiming to achieve high scientific and practical significance.

1.2 Motivations and contributions

Because of the high practical need of solving the above problems, many research studies have been conducted taking into account the multi coverage and connectivity constraints. However, these works mainly focus on extending the network lifetime or scheduling the sensor set with predefined locations of all the nodes. Moreover, when solving the Q-connectivity problem, most of the works address

Figure 1.1: WSN illustration.

the multi-path connectivity among the sensors without considering the full paths from the targets to the base station, which directly serves the purpose of target-centric WSNs. While a small number of researches have been done solving multi coverage and multi connectivity at the same time [20], [23], these works only consider the ideal 2D environment, which can not be yielded in practice. Besides, the available researches in 3D environment only address the 1-Coverage problem [21] or Q-Coverage with fixed number of sensors [22]. Hence, our work is the first to tackle the Q-Coverage and Q-Connectivity problems concurrently with the aim of minimizing the number of nodes. The contribution of this thesis is as follows:

- Formulating the node minimization problem under Q-Coverage and Q-Connectivity constraints in 3D environment.
- Proposing a two-phase graph-based solution with superior performance compared to the state of the art algorithms.
- Providing a dataset of two distinct levels of height variation in 3D environment for the problem.

1.3 Thesis organization

The remaining sections of the thesis are structured in the following manner.

- **Chapter 2** provides essential preliminary concepts required for the thesis.
- **Chapter 3** formulates the problem, introduces its application, and discusses related works.
- **Chapter 4** presents our proposed method to solve the problem.
- **Chapter 5** elaborates the details of our extensive experiments.

CHAPTER 2. PRELIMINARIES

This chapter elucidates the essential theoretical framework required for the investigation and resolution of the problem posited in the thesis. It encompasses three distinct sections, regarding the rudimentary principles of optimization problems, the principles of graph theory, and ultimately, the foundational concepts of WSN. The initial section introduces and expounds upon two distinct categories of optimization problems. Subsequently, the second section provides comprehensive explanations of graph definitions and expounds upon the MST and clique partitioning problem. Finally, the chapter delves into an extensive examination of WSN, encompassing their definition, structure, applications, and associated challenges.

2.1 Optimization problem

Optimization problems lie at the heart of countless disciplines and industries, seeking to maximize or minimize a certain objective within given constraints. Whether in mathematics, engineering, economics, or computer science, the essence of an optimization problem revolves around finding the best possible solution among a set of alternatives. These problems can be complex, involving multiple variables, constraints, and interdependencies, requiring the application of various algorithms and techniques. Optimization encompasses a wide range of scenarios, from finding the most efficient route for a delivery truck to determining optimal resource allocation in a manufacturing process.

Multiple optimization problems have been proven to belong to the class of NP-hard problems. For these problems, it is currently infeasible for humans to find an exact solution within polynomial time. Consequently, researchers often approach these problems through approximate solving methods utilizing various techniques, such as the Monte Carlo method, heuristic algorithms, and so forth.

Classification of optimization problems into distinct groups can be approached from several perspectives. A common approach involves categorizing problems into two classes based on the nature of the variables: continuous or discrete, as follows:

- **Continuous optimization:** a branch of optimization that deals with optimizing continuous mathematical functions and continuous variables.
- **Combinatorial optimization:** the type of optimization which handles discrete variables, such as integer, permutation, etc.

2.1.1 Continuous optimization problem

The standard formulation of continuous problem can be expressed as:

$$\begin{aligned} \textbf{Minimize} \quad & f(x). \\ \textbf{Subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & h_j(x) = 0, \quad j = 1, \ldots, p, \end{aligned}$$

where

- $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function.
- $g_i(x) \leq 0$ and $h_j(x) = 0$ are inequality and equality constraints, respectively.
- $m, p \in \mathbb{N}$ are the the number of inequality and equality constraints, respectively.

2.1.2 Combinatorial optimization problem

Formally, a combinatorial optimization problem A is a quadruple (I, f, m, g) [24]–[26], where:

- I is a set of instances.
- Given an instance $x \in I$, $f(x)$ is the set of feasible solutions.
- Given an instance x and a feasible solution $y \in f(x)$, $m(x, y)$ denote the measure of y.
- g is the objective function, and is either **min** or **max**.

The goal is then to find for some instance x an optimal solution, that is, a feasible solution y with $m(x, y) = g\{m(x, y')|y' \in f(x)\}$.

2.2 Wireless Sensor Network preliminary

Within this section, the core concepts of WSN are presented, encompassing their fundamental definitions, categorization, applications, and associated challenges. Additionally, an exploration is undertaken regarding the coverage and connectivity issues that arise within the context of WSNs.

2.2.1 Definitions

a, Sensor definition

According to [6], in a broad sense, a sensor can be described as a mechanism that detects and reacts to various forms of physical stimuli, including heat, light, sound, pressure, magnetism, and more. It converts the magnitude or characteristic of the physical stimulus into measurable signals, such as electrical or mechanical

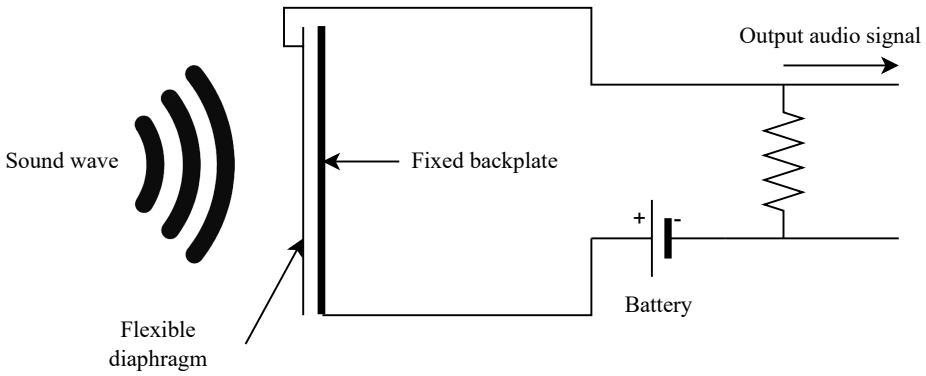


Figure 2.1: An acoustic sensor–condenser microphone [6].

signals [27]. These signals are typically converted into digital format to generate data that can be interpreted and used for sensing purposes. Sensors serve as a crucial link between the physical realm and the realm of electrical devices, like computers, enabling individuals to comprehend, monitor, and manage machinery and environments.

In the realm of scientific applications, sensors exhibit a vast array of variations and forms, capable of detecting nearly every imaginable type of physical stimulus. Traditionally, these sensors can be categorized into various types based on the specific quantity they aim to measure [27], [28] as follows:

- *Mechanical quantities* encompass parameters like position, displacement, velocity, acceleration, vibration, force, torque, pressure, etc.
- *Fluid quantities* include properties such as density, viscosity, thickness, surface morphology, etc.
- *Electrical quantities* involve measurements of voltage, current, impedance, resistance, conductance, electric field, magnetic field, polarization, etc.
- *Thermal quantities* cover measurements related to temperature, infrared imaging, and related aspects.
- *Chemical quantities* involve measurements of concentration, pH values, enzymes, ions, etc.
- *Biological quantities* encompass measurements of concentrations of enzyme substrates, antigens, antibodies, etc.

Example [6] The diagram in Figure 2.1 illustrates the operational principle of an acoustic sensor, specifically a condenser microphone. This type of microphone converts sound signals into electrical signals. The microphone consists of a flexible diaphragm and a fixed back plate, which together form a parallel-plate capacitor.

When sound waves reach the condenser microphone, they exert pressure on the diaphragm, causing it to vibrate in response. As a result, the distance between the flexible diaphragm and the fixed back plate changes, leading to a variation in the electrical charge on the plates and generating output electrical signals.

In the provided figure, the electronic circuit accompanying the condenser microphone is a resistor-capacitor (RC) circuit. It supplies a constant charge, denoted as Q, to the capacitor. The capacitance (C) of the capacitor is inversely proportional to the distance between the two plates. The capacitance equation is given by $C = Q/V$, where Q represents the charge in coulombs, C denotes the capacitance in farads, and V represents the potential difference in volts.

When the diaphragm vibrates, causing a change in the capacitance C, the voltage across the capacitor also changes. This voltage change is observed across the series resistor in the circuit. The voltage across the resistor can be amplified for subsequent processing or recording purposes.

b, Sensor node definition

The fundamental component of a sensor network is a sensor node, which not only incorporates sensors for perceiving the physical environment but also encompasses additional modules for processing and transmitting sensory data. Figure 2.2 illustrates a typical architecture of a sensor node, comprising the following units: the sensing unit, communication unit, processing unit, and power unit. Depending on the specific application and requirements, a sensor node may also integrate supplementary units, such as a locomotive unit for enabling movement, an power generator for extracting energy from the surroundings, a GPS unit for determining geographical location, etc [6], [29]. The details of those units are presented below:

- The *sensing unit* within a sensor node can encompass multiple types of sensors and typically includes analog-to-digital converters (ADCs). These converters are responsible for converting analog signals generated by the sensors into digital signals, which are then processed further by the processor.
- The *processing unit*, often accompanied by a small memory, handles the necessary procedures for the sensor node to collaborate with other nodes in executing assigned sensing tasks.
- The *communication unit* facilitates the exchange of data between individual nodes. In the case of wireless communication, sensor nodes are equipped with transceivers, enabling them to form an ad-hoc WSN autonomously. A transceiver incorporates essential circuitry such as a transmitter, receiver, modu-

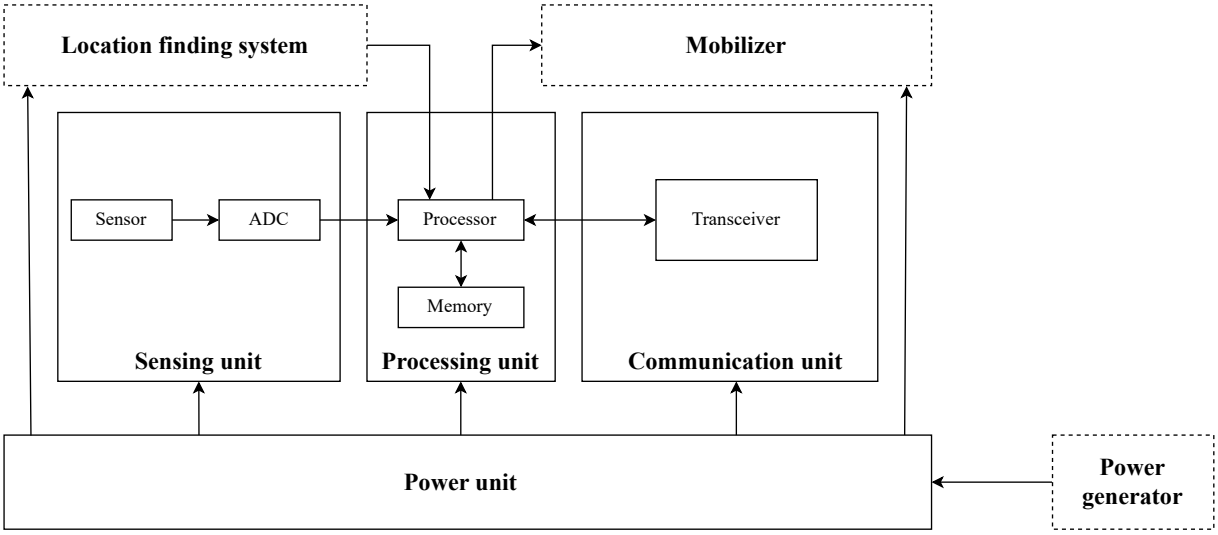


Figure 2.2: Sensor node architecture.

lator, demodulator, power amplifiers, filters, antenna, and more, converting data streams to radio waves and vice versa.

- The *power unit* holds significant importance within a sensor node, and power sources may be supplemented by power scavenging units, such as solar cells. Additionally, there may be other application-dependent subunits incorporated within the sensor node.

c, WSN definition

Sensor nodes are usually deployed in a field of interests to monitor some physical phenomena. Such a field is called a *sensor field*, and the sensor nodes form a *sensor network* [6]. When the sensor nodes are wireless devices, we obtain a *wireless sensor network*. The integration of sensor nodes and wireless communication technologies in WSNs revolutionizes data collection, analysis, and control systems, enabling efficient monitoring and management of physical environments in a variety of applications. The next part elaborates on several sensor network architectures and their applications.

Several sensor network architectures A sensor network normally consists of a large number of sensor nodes and one or more sinks/base stations. Sensor nodes are responsible for monitoring physical phenomena and generating collected sensory data. A sink acts as a gateway between a sensor networks and other networks, or between the sensor network and human operations. These sinks also serve as the network's central controller as they possess better processing capabilities and abundant energy sources. In some cases, both sinks and sensors are attached with locomotives enabling them to move within the sensor field, which enhances the efficiency of performing tasks. In some literature, sink is referred to as base station.

Sensor networks can be categorized into the following typical architectures [6]:

- **Homogeneous and Heterogeneous Networks:**
  - Homogeneous networks consist of sensor nodes with the same sensing, processing, communication, and other capabilities (excluding the initial power supply). Figures 2.3(b) and (d) illustrate two homogeneous networks.
  - Heterogeneous networks include sensor nodes of different capabilities, such as different sensing radius. Homogeneous networks are more commonly applied because they are used to monitor multiple objects in the domain with the same objective. Figures 2.3(a) and (c) show two heterogeneous networks.
- **Single-Hop and Multi-Hop Networks:**
  - Single-hop networks enable direct communication between sensor nodes and the sink without any intermediaries. This can be costly, especially for a large-scale network. Figure 2.3(a) illustrates a single-hop network.
  - Multi-hop networks allow some sensor nodes to use a multi-hop path consisting of other nodes as relays to deliver their data to the sink instead of transmitting to the sink directly. Multi-hop networks are commonly used due to their flexibility and applicability in large observation areas. Figure 2.3(b), (c) presents multi-hop networks.
- **Stationary and Mobile Networks:**
  - Stationary sensor networks are deployed with fixed sensor nodes that do not move during the monitoring process.
  - Mobile sensor networks involve all sensor nodes attached to devices capable of moving within the observed area to collect data on various observed objects. Mobile sensor networks offer flexibility and fault-tolerance but are expensive to deploy. Therefore, a common approach is to combine mobile sensor nodes with stationary sensor nodes, which forms a hybrid network, to enhance network efficiency. In some cases, a sink can also be a mobile node, and it moves around to collect sensing data, as shown in Figure 2.3(d).

Applications of WSN

WSNs have gained significant attention due to their diverse range of applications across various fields. The applications of WSNs are vast and encompass several

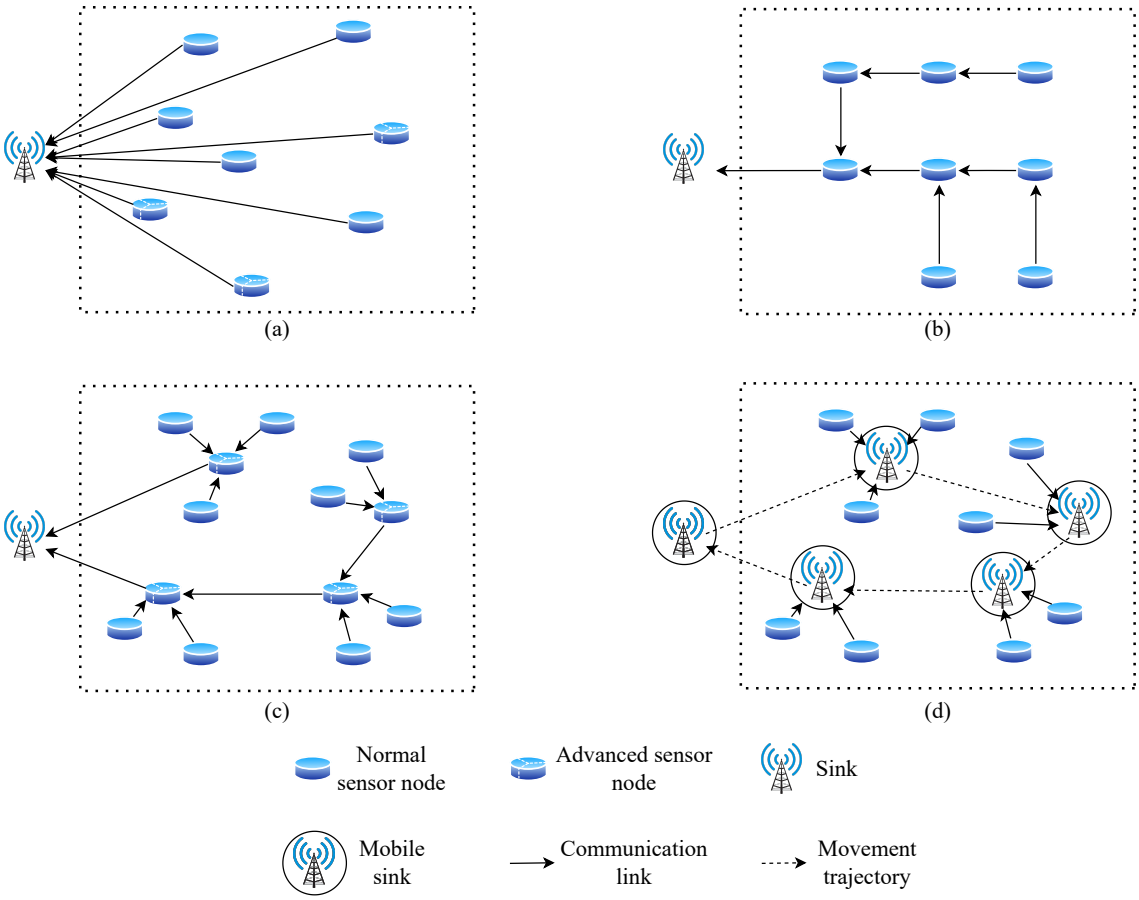


Figure 2.3: Sensor network scenarios: **(a)** single-hop, heterogenous, stationary network; **(b)** multi-hop, homogeneous, stationary network; **(c)** multi-hop, heterogeneous, stationary network; **(d)** single-hop, homogeneous, stationary network with a mobile sink.

domains, each benefiting from the capabilities and functionalities offered by these networks [30]–[34]. Some of the applications of WSNs are listed below:

- **Environmental Monitoring [35]–[37]:** WSNs enable the collection and analysis of data related to air quality, temperature, humidity, and pollution levels. This information aids in the detection of natural disasters, environmental conservation efforts, and ensuring the overall well-being of ecosystems.
- **Healthcare [38]–[40]:** WSNs facilitate patient monitoring by allowing continuous and real-time tracking of vital signs such as heart rate, blood pressure, and oxygen levels. These networks enhance the quality of care, enable remote patient monitoring, and ensure timely medical interventions.
- **Agriculture [41]–[43]:** WSNs find applications in agriculture by monitoring soil moisture, temperature, and humidity levels. By providing real-time data, WSNs enable optimized irrigation systems, efficient resource allocation, and improved crop yield while conserving water resources.
- **Industrial Automation [44]–[46]:** WSNs play a vital role in industrial auto-

mation by providing real-time monitoring of machinery, detecting faults, and optimizing operational efficiency. These networks enable predictive maintenance, reducing downtime, and enhancing productivity in manufacturing processes.

- **Smart Homes and Buildings [47]–[49]:** WSNs offer intelligent solutions for energy management, security, and automation in smart homes and buildings. These networks enable remote control of appliances, efficient energy usage, enhanced safety through intrusion detection systems, and improved comfort for occupants.
- **Transportation Systems [50]–[52]:** WSNs contribute to transportation systems by enabling traffic monitoring, congestion detection, and vehicle tracking. These networks facilitate real-time traffic management, optimized traffic flow, and improved road safety through early incident detection.
- **Military and Surveillance [53]–[55]:** WSNs provide reliable and covert communication, surveillance of sensitive areas, and intrusion detection in military and security applications. These networks enhance situational awareness, aid in border surveillance, and ensure national security.
- **Structural Health Monitoring [56]–[58]:** WSNs aid in the assessment and analysis of the structural integrity of bridges, buildings, and dams. These networks monitor structural parameters such as vibration, strain, and temperature, enabling early detection of potential failures and ensuring public safety.

In conclusion, wireless sensor networks offer a wide range of applications across various sectors, contributing to improved efficiency, safety, and sustainability in diverse domains.

2.2.2 Challenges in Wireless Sensor Network

As mentioned above, sensor networks hold great promise for numerous applications in various environmental conditions. However, deploying and maintaining a WSN still pose significant challenges such as energy consumption, connectivity, fault tolerance, etc. According to [6], the following are some challenges in WSNs:

- **Energy Efficiency:** In WSNs, it is crucial to ensure that sensor nodes remain operational. However, most sensor nodes have limited energy resources, necessitating the efficient minimization of energy consumption. Several energy-saving mechanisms have been implemented in node hardware, such as using different operating modes with varying energy consumption rates. However, scheduling the hardware operations should be coordinated and executed

network-wide to achieve optimal performance.

- **Network Autonomy:** Sensor nodes can either be placed deterministically or scattered randomly across a field of interest. In certain remote or hazardous environments, random scattering of nodes might be the only viable approach for deploying a sensor network. In such scenarios, these dispersed nodes should possess the capability to self-organize, forming an autonomous network that autonomously determines the structure and topology of the network. An autonomous sensor network must exhibit the ability to independently schedule sensing tasks and establish optimal delivery routes. Furthermore, it is crucial for the network to continuously monitor its own health and status, and adapt its operational parameters accordingly, depending on the prevailing conditions. In certain instances, a sensor network may need to interact with external maintenance mechanisms or external network interfaces to enhance its overall operational efficiency. The integration of such external interactions can further improve the network's performance and effectiveness.
- **Fault Tolerance:** In many cases, operating in harsh environments can cause certain positions in the network to become faulty, leading to reduced or non-functional network operations. In such cases, these networks need to organize backup paths or nodes to avoid such situations caused by node failures. Additionally, addressing transmission issues and information loss in sensor networks is crucial. Having multiple redundant paths can help ensure stable network operations.
- **Network Scalability:** In many applications, the number of sensors in a WSN can reach thousands, tens of thousands, or even millions. In such large-scale networks, scalability is a critical factor to ensure that the network's performance is not significantly degraded as the network size increases. Algorithms and protocols designed for small-scale networks do not necessarily perform well in large-scale networks. Achieving network scalability requires prioritizing distributed algorithms and self-governance.
- **Data Accuracy:** The primary objective of sensor networks is to acquire precise and reliable information. To enhance accuracy, cooperative sensors engage in joint signal processing. Nevertheless, a challenge arises where maintaining high information accuracy clashes with the goal of prolonging the network's lifespan. This is due to the increased usage of sensors for various tasks, including sensing, processing, and data delivery, which leads to higher energy consumption. Thus, it becomes essential to strike a favorable balance

between data accuracy and energy consumption. Finding this balance ensures that the network operates optimally by providing accurate information while efficiently managing energy resources. By achieving this equilibrium, sensor networks can fulfill their intended tasks effectively without compromising the overall network longevity.

- **Information Security:** Information security is a fundamental and widespread necessity in nearly all types of networks, and sensor networks are no exception. To ensure information security, it is crucial that sensing data is accessed, transmitted, and processed in a secure and private manner. However, typical security algorithms tend to be resource-intensive, which poses a challenge in the context of resource-constrained sensor networks. In such scenarios, security algorithms must be tailored and optimized to accommodate the limitations of sensor networks. These modifications are essential to adapt the algorithms to function efficiently within the resource-constrained environment of sensor networks while still upholding the required level of security. By striking a balance between security and resource utilization, sensor networks can maintain the confidentiality, integrity, and availability of their data without overburdening their limited resources.

  In relation to the challenges mentioned earlier, the thesis focuses on addressing the challenge of fault tolerance in the network by utilizing heuristic algorithms to create multiple pathways from a specific target to the base station and employ multiple sensors to monitor each target while minimizing the total number of nodes. The next sections elaborate on the coverage and connectivity problem in WSN.

2.2.3 Coverage in Wireless Sensor Network

This section discusses sensor coverage models used to analyze the coverage capability of sensors in a given space as well as highlights three classes of coverage problems in Wireless Sensor Networks (WSNs): point coverage, area coverage, and barrier coverage.

a, Sensor coverage models

A sensor coverage model refers to a mathematical or computational representation used to assess and analyze the coverage capability provided by a sensor at a given space point. It determines the extent to which the deployed sensor can effectively monitor and detect events or phenomena of interest at a specific point. The coverage model is usually represented as a function $f : (d, \phi) \rightarrow \mathbb{R}^+$ that takes into account factors such as Euclidean distance d (and the angle ϕ) between the sensor

and the point. The output of that function is named *coverage measure* of the given point, which is a non-negative number. According to [6], there are four basic sensor coverage models as follows:

- **Boolean Sector Coverage Models:** The sector coverage model, also known as the Boolean sector model, is inspired by directional cameras and serves as a Boolean directional coverage model [59] (see Figure 2.4a). The coverage function of the sector model, denoted as $f(d(s,z),\phi(s,z))$, is defined in Equation 2.1:

$$f(d(s,z),\phi(s,z)) = \begin{cases} 1 & \text{if } d(s,z) \leq r_s \text{ and } \phi_s \leq \phi(s,z) \leq \phi_s + \omega, \\ 0 & \text{otherwise,} \end{cases} \tag{2.1}$$

  where $d(s,z)$ and $\phi(s,z)$ is the Euclidean distance and the angle between sensor s and point z, respectively; ϕ_s is called an orientational angle, ω is called the visual angle of the sector model, and r_s is called the sensing range. This function defines a sector centered at s which has a radius r_s. If a point z lies inside that sector, it is said to be covered by the sensor and vice versa.

- **Boolean Disk Coverage Models:** The Boolean Disk Coverage Model (BDCM) is the most widely utilized coverage model in the literature because of its simplicity (see Figure 2.4b). The coverage model is formulated as:

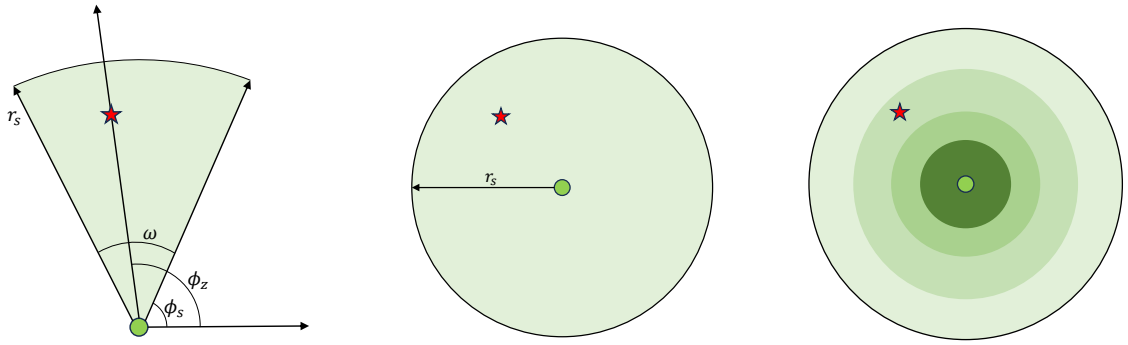
$$f(d(s,z)) = \begin{cases} 1 & \text{if } d(s,z) \leq r_s, \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

  where $d(s,z)$ is the Euclidean distance between sensor s and point z and r_s is the sensing range of s. This function defines a disk which centres at s and has radius r_s. If z lies inside the disk, z is said to be covered by s and vice versa.

- **Attenuated Disk Coverage Models:** Some researchers claim that as the distance from the sensor increases, the sensing quality of a sensor diminishes [60], [61]. To represent this attenuation in sensing quality, an attenuated disk coverage model is employed. An instance of such an attenuated disk coverage model is formulated as follows.

$$f(d(s,z)) = Cd^{-\alpha}(s,z), \tag{2.3}$$

  where $d(s,z)$ is the Euclidean distance between sensor s and point z, C is a constant which depends on the sensor configuration. This function indicates that the closer the point be to the sensor, the larger the coverage measure is.



(a) Boolean Sector Coverage Model.

(b) Boolean Disk Coverage Model.

(c) Truncated Attenuated Disk Model.

Figure 2.4: Illustrations of sensor coverage models.

- **Truncated Attenuated Disk Models:** When employing the attenuated disk coverage model, as the distance between a sensor and a specific point in space increases significantly, the coverage measure decreases to a negligible level (see Figure 2.4c). In such instances, the coverage measure can be simplified by truncating it for larger distances. For instance, Zou et al. [62] introduce a truncated attenuated coverage function to address this. It is formulated as:

$$f(d(s,z)) = \begin{cases} Cd^{-\alpha}(s,z) & \text{if } d(s,z) \leq r_s, \\ 0 & \text{otherwise,} \end{cases} \tag{2.4}$$

  where $d(s,z)$ is the Euclidean distance between sensor s and point z, C is a constant depending on the sensor configuration, and r_s is the sensing range of s.

b, WSN coverage problems

According to [6], coverage problems can be classified into three classes, which are point coverage, area coverage and barrier coverage. Those problems are elaborated as follows:

- **Point Coverage Problems:** Within this category of problems, the primary objective is to ensure coverage of a discrete set of points, which may represent physical targets in practical applications (see Figure 2.5a). This class can be further divided into two distinct categories: *node placement optimization*, which involves identifying optimal sensor locations to minimize network cost, and *coverage lifetime optimization*, which focuses on scheduling sensors to maximize the network's operational lifespan. In point coverage problem, there are three types of coverage constraint which are often considered: (1) 1-Coverage, where each target is covered by 1 sensor; (2) K-Coverage, where

each target is covered simultaneously by K sensors and Q-Coverage, where target t_i is covered by q_i sensors simultaneously (i.e., the coverage priority varies among different targets). K and Q- Coverage can ensure fault tolerance of the network as the physical stimuli from a target can still be sensed when one or more sensors are down. In this thesis, Q-Coverage constraint is investigated.

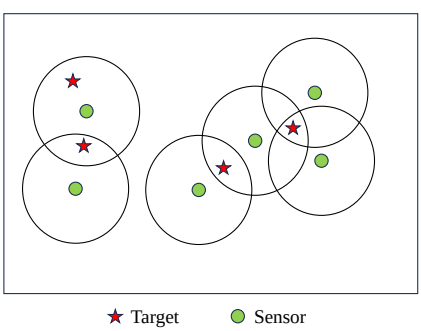
- **Area Coverage Problems:** Differing from point coverage problems, area coverage problems require complete coverage of an entire designated area, ensuring that every point within the area is covered (see Figure 2.5b). This class can be further classified into several categories: *critical sensor density problem*, which involves determining the minimum number of sensors required per area unit to achieve full coverage; *sensor activity scheduling problem*, where sensor scheduling is optimized to maximize the network's operational lifespan; and *node movement strategy problem*, which explores the utilization of mobile nodes to control network coverage while minimizing costs associated with both product and movement.
- **Barrier Coverage Problems:** The goal of barrier coverage problems is to identify specific desired coverage attributes, if they exist within the network (see Figure 2.5c). This class consists of two sub-problems: *building intrusion barriers*, which involve the detection of any trespassing by a mobile object within the sensor field, and *finding penetration paths*, where every point satisfies the required coverage measure along designated paths.

In conclusion, the coverage problem in WSN is an important issue and has wide range applications in real-life scenarios.

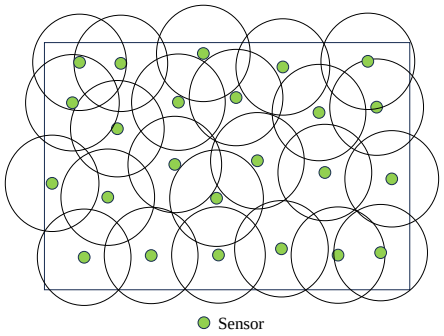
2.2.4 Connectivity in Wireless Sensor Network

Connectivity is a critical issue in the design and control of WSNs [6]. It involves ensuring that all sensor nodes can establish connections with base stations or sink nodes. While this task is typically handled by the network layer, it can also be integrated into coverage control as a multi-layer approach.

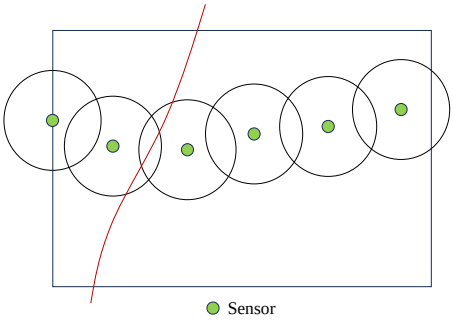
In WSNs, wireless sensors communicate via wireless transceivers. Two sensors can directly connect if they can transmit and receive data through the radio channel. Alternatively, nodes can be connected through multi-hop transmission, where intermediate nodes act as relays. A network is considered fully connected if data can be transmitted from any sensor node to other nodes or the sink node, using both single-hop and multi-hop transmissions. In a fault-tolerant network, there can be more than one path to transmit data between nodes [12]–[14], [63]–[67], or



(a) Point Coverage.



(b) Area Coverage.



(c) Barrier Coverage.

Figure 2.5: Illustrations of WSN coverage problems.

from a target to the base station [20], [23]. Such number of path can be fixed by a number K or varies between targets by a priority vector Q. Those define K and Q-Connectivity constraint, respectively. In this thesis, Q-Connectivity constraint is investigated.

To illustrate, Figure 2.6 depicts a Q-connected network example with both single-hop and multi-hop connections from sensor nodes to the base station. There are two targets with 2 and 3 connection paths to the base station. The disk model is commonly used, where a node can communicate with others within a circle area centered at its position. The radius of the disk is called communication range. Various connectivity models also consider factors such as variable transmission power and transmission errors. For the sake of simplicity, this thesis only considers the communication range.

It's important to note that the transmission unit is independent of the sensing unit within a sensor node, and the communication range can be adjusted at different distances. Additionally, network connectivity design is closely related to coverage in wireless sensor networks and can be combined with operational scheduling.

Overall, ensuring connectivity in WSNs is a significant and necessary consideration that requires attention and resolution.

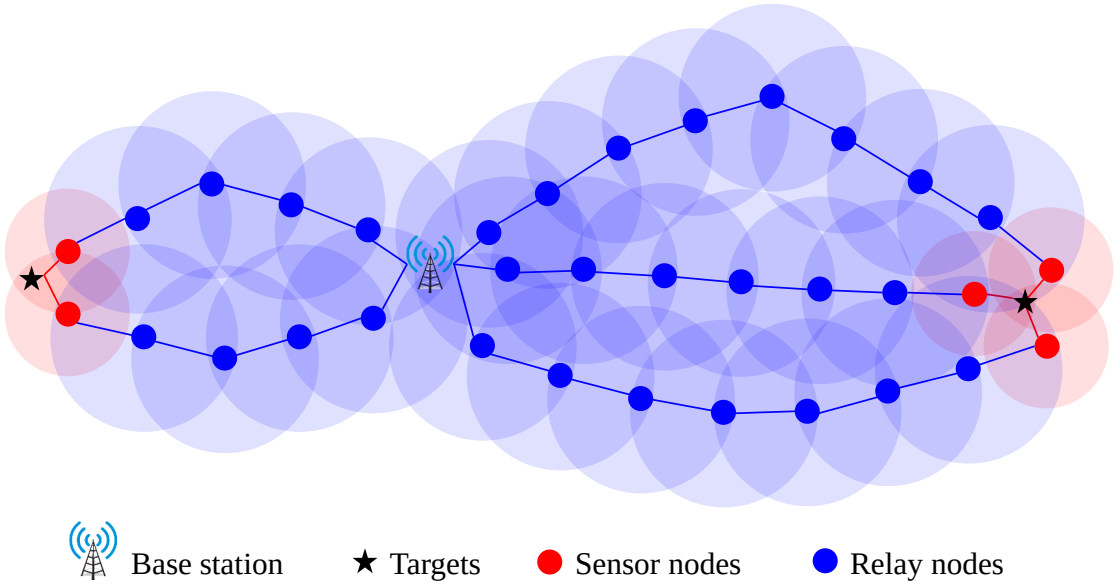


Figure 2.6: An example of Q-Connectivity WSN.

2.3 Graph preliminary

A graph is a fundamental data structure used to represent relationships between objects or entities. It consists of a set of vertices or nodes, connected by edges or arcs. Graphs are widely used in various fields, including computer science, mathematics, social sciences, and engineering.

In a graph, the vertices represent the entities or elements, while the edges denote the connections or relationships between them. These connections can be either directed or undirected, depending on whether the edges have a specific direction or not. For example, in a social network graph, the vertices can represent individuals, and the edges can represent friendships or connections between them.

This section covers some fundamental definitions related to graph, including graph types, path in graph, tree, and minimum spanning tree (MST). All of the definitions are referenced from [24].

2.3.1 Graph definitions

Based on the attributes of graph edges, we can categorize graphs into five types as follows.

Definition 1. *A simple undirected graph $G = (V, E)$ consists of a set of vertices V and a set of edges E. Each edge in E is an unordered pair of two distinct elements from V (see Figure 2.7a).*

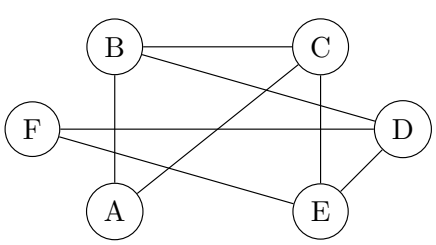
Definition 2. *A undirected multigraph $G = (V, E)$ consists of a set of vertices V and a collection of edges E. Each edge in E is an unordered pair of two distinct*

elements from V. Two edges that correspond to the same pair of vertices are called parallel edges (see Figure 2.7b).

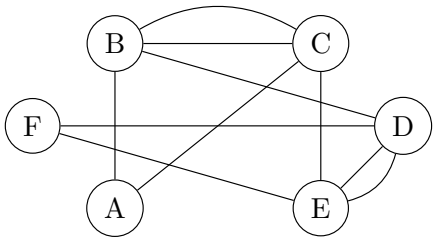
Definition 3. *A pseudo undirected graph $G = (V, E)$ consists of a set of vertices V and a collection of edges E. Each edge in E is an unordered pair of two elements, not necessarily distinct, from V. An edge is called a loop if it consists of a pair of identical vertices (see Figure 2.7c).*

Definition 4. *A simple directed graph $G = (V, E)$ consists of a set of vertices V and a set of edges E. Each edge in E is an ordered pair of two distinct elements from V (see Figure 2.7d).*

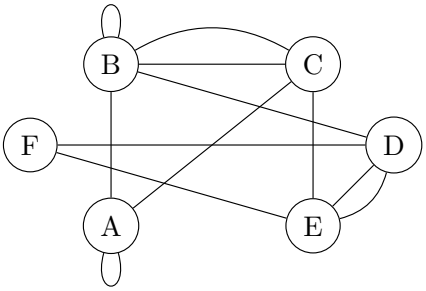
Definition 5. *A directed multigraph $G = (V, E)$ consists of a set of vertices V and a collection of edges E. Each edge in E is an ordered pair of two distinct elements from V. Two edges that correspond to the same pair of vertices are called parallel edges (see Figure 2.7e).*



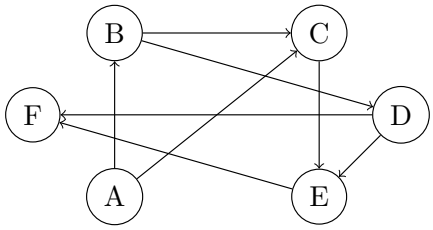
(a) Simple undirected graph.



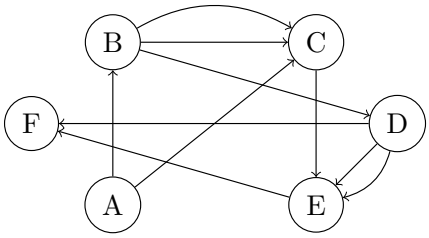
(b) Undirected multigraph.



(c) Pseudo undirected graph.



(d) Simple directed graph.



(e) Directed multigraph.

Figure 2.7: Some types of graph.

After presenting about some types of graph, some other necessary definitions about path and connected component in graph are introduced below.

2.3.2 Path and connected component in graph

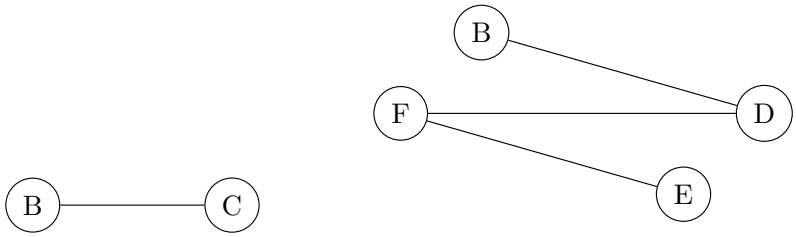
Definition 6. *In an undirected graph G, two vertices u and v are adjacent if there exists an edge $e = (u, v)$ in the graph. We say that the edge (u, v) is incident to both vertices u and v, or that it connects vertex u and vertex v. Vertex u and vertex v are two endpoints of the edge.*

Figure 2.8a presents an edge with two endpoints B and C.

Definition 7. *In an undirected graph $G = (V, E)$, a path of length n from vertex u to vertex v is a sequence: $x_0, x_1, \ldots, x_n$, where the starting vertex is $u = x_0$, the ending vertex is $v = x_n$, and $(x_i, x_{i+1}) \in E$ for $i = 0, 1, \ldots, n - 1$.*

A path corresponds to a sequence of $n-1$ edges: $(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)$. A path with the same starting and ending vertex ($u = v$) is called a cycle. A path or cycle is said to be simple if no vertex is repeated along the path.

Figure 2.8b illustrates a path of length 3 with the vertex sequence (B, D, F, E).



(a) Edge. **(b)** Path.

Figure 2.8: Edge and path in graph.

Definition 8. *An undirected graph $G(V, E)$ is considered connected if there exists a path between any two arbitrary vertices of the graph.*

Definition 9. *Graph $H(W, F)$ is a subgraph of $G(V, E)$ if $W \subseteq V$ and $F \subseteq E$.*

If G is not a connected graph, it can be partitioned into multiples connected subgraph. Each such subgraph is called a connected component of G.

Figure 2.9 shows an example of a graph with two connected components. We

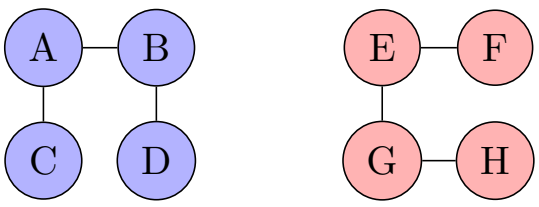


Figure 2.9: A graph with 2 connected components, colored in blue and red.

can easily find all connected components of a graph using BFS (Breadth-first search) or DFS (Depth-first search). An approach using DFS is presented in Algorithm 1. Since each vertex is visited once and each edge is visited twice and all other operations take $O(1)$, the total complexity of the algorithm is $O(|V| + |E|)$.

Algorithm 1: Finding all connected components using DFS

Input : $G(V, E)$: An undirected graph.
Output: All connected components.

```
component_id = {}
current_id = 0
for k ∈ V do
    if Visited[k] == False then
        DFS(V, k, current_id, component_id)
        current_id+ = 1
    end
end
return component_id
```

Algorithm 2: DFS($V, k, current\_id, component\_id$)

```
component_id[k] = current_id
Visited[k] == True
for p ∈ V.adj[k] do
    if Visited[p] == False then
        DFS(V, p, current_id, component_id)
    end
end
```

The fundamental concepts of path and connected component in graph theory, as defined above, serve as the underlying principles for establishing connectivity in WSN. These definitions provide a solid foundation that enables the application of graph algorithms to address the problem presented in this thesis.

2.3.3 Spanning tree

Within the domain of graph theory as well as data structures and algorithms, the notion of a tree holds paramount importance. In the context of WSN, the utilization of tree-like graph structures prevails as a prevalent and esteemed approach. Constructing these trees serves a dual purpose: guaranteeing the connectivity of all network nodes while simultaneously minimizing costs through the creation of minimal spanning trees. This section entails the introduction of the definitions of tree and spanning tree. Subsequently, the MST problem is presented, accompanied by an exposition of Prim's algorithm as a means to solve it.

Definition 10. *A tree is a connected acyclic undirected graph. A forest is a acyclic*

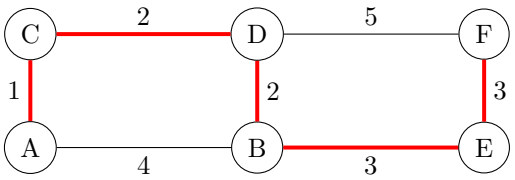


Figure 2.10: A graph $G(V, E)$ along with its MST colored in red.

graph.

Tree characteristics: Let T be a graph with $n \geq 2$ vertices. The following propositions are equivalent:

- T is a tree.
- T is connected and has $n - 1$ edges.
- T is acyclic and has $n - 1$ edges.
- T is connected and each edge is a bridge.
- Between any two vertices of T, there exists a unique simple path without cycles.
- T is acyclic, but the addition of any new edge results in a unique cycle.

Definition 11. *Given a connected undirected graph $G(V, E)$, a tree $T(V, F)$ with $F \subset E$ is a spanning tree of G.*

With the above definitions, the MST problem can be formulated as below:

Given a connected undirected graph $G(V, E)$. Each edge of G is assigned a real number $c(e)$ which is its length. Suppose that $H(V, T)$ is a spanning tree of G. Let $c(H)$ denote the length of H, which is calculated as:

$$c(H) = \sum_{e \in T} c(e). \tag{2.5}$$

Let S denote the set of all spanning tree of G. The objective of MST is:

$$\underset{H \in S}{\text{minimize}} \quad c(H).$$

An example of MST is illustrated in Figure 2.10.

Prim's algorithm is a widely-used greedy algorithm for finding the minimum spanning tree $H(V, T)$ in a connected, weighted undirected graph $G(V, E)$ with n vertices. It falls under a class of algorithms called greedy algorithms that find the local optimum in the hopes of finding a global optimum. It starts from one vertex and keep adding edges with the lowest weight until we reach our goal. The steps

for implementing Prim's algorithm are as follows:

- Initialize the minimum spanning tree with a vertex chosen at random.
- Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree.
- Keep repeating step 2 until we get a minimum spanning tree (i.e., all the vertexes of the graph are in the tree).

Since the algorithm only compares limited number of edges per loop, it can work well on dense graph, where there are larger number of edges compared to vertexes. The pseudo code of Prim's algorithm is given in Algorithm 3.

Algorithm 3: Prim's Algorithm for MST

Input : $G(V, E)$: A connected, weighted undirected graph.
Output: The set of edges in a MST.
$T = \emptyset$
$U = \{1\}$
while $|T| < n - 1$ **do**
  Find the lowest cost edge (u, v) with $u \in U$ and $v \in V \backslash U$
  $T = T \cup \{(u, v)\}$
  $U = U \cup \{v\}$
end
Return T.

When being implemented with adjacency list and Fibonacci heap, the time complexity of Prim's algorithm is $O(|E| + |V|log|V|)$.

2.3.4 Clique-partitioning

In this section, the definition of clique and clique-partitioning are introduced. Subsequently, an algorithm for clique-partitioning a graph are introduced.

Firstly, the definition of clique is as follows:

Definition 12. *Given a simple graph $G(V, E)$, a subgraph $H(W, F)$ of G is a clique in G if H is a complete graph.*

After having the definition of clique, the clique-partitioning problem is introduced below.

Definition 13. *The problem of clique-partitioning a graph involves determining the minimum number of cliques needed to ensure that each vertex in the graph is included in exactly one clique.*

The graph coloring problem entails identifying the minimum set of "colors" required to assign a distinct color to each vertex in a graph, while ensuring that

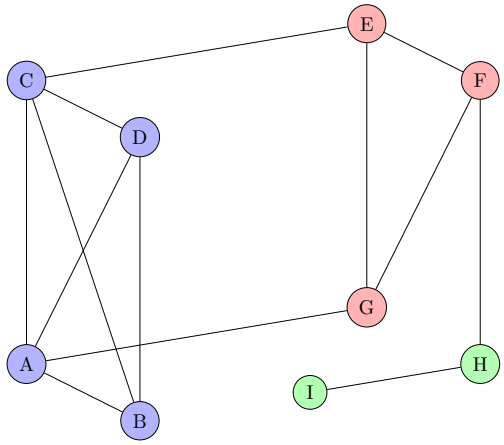


Figure 2.11: An example of clique-partitioning in a graph. There are 3 cliques which are colored in red, blue and green.

connected vertices are not assigned the same color. The clique-partitioning problem in a graph G can be viewed as an equivalent problem of coloring the complement graph G' (see Figure 2.11). Therefore, vertex-coloring algorithms can also be utilized to solve the clique-partition problem.

There are various algorithms studied to solve the problem [68]–[75]. In [75], the authors presented a novel algorithms which outperformed the previous algorithms in the clique-partitioning task. Hence, it is utilized later in this thesis. The detail of the algorithm is presented in Algorithm 4. The complexity of it is $O(n^2)$ with n being the number of vertexes in the graph, as mentioned in [75].

Algorithm 4: Clique partition algorithm [75]

Input : Graph $G(V, E)$.
Output: A clique-partition of the graph.
If all vertices have a degree of zero, proceed to Step 7.
Select a vertex with the smallest non-zero degree and denote it as N_1.
From all vertices connected to N_1, choose the one with the smallest degree and label it as N_2.
Combine N_1 and N_2 into a new composite vertex called N_3.
Update the graph by replacing vertices N_1 and N_2 with the new vertex N_3. In the modified graph, a vertex N_x is connected to N_3 if and only if N_x was previously connected to both N_1 and N_2. The degrees of the affected vertices are adjusted accordingly.
Return to Step 1.
Any vertices that have not been grouped together in clusters are treated as individual clusters.

CHAPTER 3. Q-COVERAGE AND Q-CONNECTIVITY PROBLEM IN 3D WIRELESS SENSOR NETWORKS

In WSNs, guaranteeing the coverage and connectivity of monitored objects is highly critical to ensure smooth data transmission and maintain network quality for effective object tracking. Additionally, considering the cost involved in building the network is essential to achieve optimal efficiency and cost-effectiveness. This chapter focuses on two significant problems in WSNs: the Q-coverage problem and the Q-connectivity problem in 3D environment. These problems present significant challenges in the development and implementation of WSNs.

3.1 Problem statement

The problem of Q-coverage and Q-connectivity in a 3D environment for Wireless Sensor Networks can be defined as follows:

Given a known 3D environment with no obstacles and specified locations of n targets and a base station, along with a vector Q where each element q_i represents the priority level of target t_i, the objective is to minimize the overall number of deployed sensor nodes and relay nodes. This deployment should satisfy two conditions: (1) target t_i must be covered by q_i sensor nodes, ensuring Q-coverage, and (2) target t_i must have q_i node-disjoint paths to the base station, ensuring Q-connectivity.

All of the notations are presented in Table 3.1.

With the above defined notations, the problem is formulated as follows: Let us consider a surveillance region A with a set of n targets, denoted as $T = \{t_i \mid i \in \{1, ..., n\}\}$, and a base station B whose locations are known. Location of target t is a triplet (t_x, t_y, t_z) where $0 \leq t_x \leq L, 0 \leq t_y \leq W, 0 \leq t_z \leq H$. The same spatial restrictions hold for the base station B, which is located at (B_x, B_y, B_z) where $0 \leq B_x \leq L, 0 \leq B_y \leq W, 0 \leq B_z \leq H$.

Additionally, we have a priority vector $Q = \{q_j \mid j \in \{1, ..., n\}\}$ associated with the targets. Within this context, each sensor node possesses a sensing range r_s and each relay node has a communication range r_c.

In this thesis, the Binary Disk Coverage Model (BDCM) is used, which means the detection probability of a sensor is 1 within the sensing range, otherwise, the probability is 0. Let $d(s, t)$ denote the Euclidean distance between sensor s and

Table 3.1: Table of notations.

Notation	***Definition***
$A(L \times W \times H)$	3D surveillance area with length L, width W, and height H
B	base station
n	number of targets
$T = \{t_i \mid i \in \{1, ..., n\}\}$	target set
m	number of sensors deployed in phase I
$S = \{s_j \mid j \in \{1, ..., m\}\}$	sensor set
p	number of relay nodes deployed in phase II
$R = \{r_k \mid k \in \{1, ..., p\}\}$	relay node set
r_s	sensing range
r_c	communication range
Q	priority vector with q_i is priority level of target t_i
$G(V, E)$	graph G with vertex set V and edge set E
O	sphere set with O_i is the sphere having center at t_i and radius r_s
I^{triad}	intersection points of three spheres
I^{pair}	midpoint of two spheres that intersect
I	$I^{triad} + I^{pair}$
p^{cover}	list of targets that $p \in I$ covers
p^{parent}	list of spheres in the triad or pair that creates p
t^{child}	list of point p such that $t \in p^{parent}$
t^{sensor}	list of sensors that cover target t
C	list of clusters

target t, then:

$$P_d(s,t) = \begin{cases} 1, & \text{if } d(s,t) \leq r_s, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

In addition, the connection between two relay nodes or between a sensor and a relay nodes is also determined by the binary disk model as follow:

$$P_c(s,s') = \begin{cases} 1, & \text{if } d(s,s') \leq r_c, \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

In order to formulate the Q-coverage and Q-connectivity constraints, we introduce binary variables b_{ij} and c_i as follows:

$$b_{ij} = \begin{cases} 1, & \text{if sensor } s_j \text{ covers target } t_i, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

$$c_i = \begin{cases} 1, & \text{if target } t_i \text{ has } q_i \text{ node-disjoint paths to } B, \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Finally, the mathematical formulation of the problem is presented as follows:

$$\textbf{Minimize} \quad |S| + |R|. \tag{3.5}$$

$$\textbf{Subject to} \quad \sum_{j=1}^{m} b_{ij} \geq q_i \qquad \forall i = 1, 2, \ldots, n, \tag{3.6}$$

$$c_i = 1 \qquad \forall i = 1, 2, \ldots, n, \tag{3.7}$$

where:

- Equation (3.5) describes the objective of the problem which is minimizing the total number of nodes.
- Equation (3.6) formulates the Q-coverage constraint, ensuring that target t_i is covered by at least q_i sensors.
- Equation (3.7) presents the formulation of the Q-connectivity constraint, which guarantees that target t_i possesses q_i node-disjoint paths leading to the base station.

In the following part, we prove that the above problem is NP-hard.

Theorem 1. *The problem of minimizing the number of nodes satisfying Q-Coverage & Q-connectivity constraint is NP-hard.*

Proof. Consider a specific scenario where we focus exclusively on the Q-Coverage problem, with the assumption that $q_i = 1$ for all $i = 1, 2, \ldots, n$, and the set $\mathcal{F}$ representing the feasible positions of sensors is finite. In this case, the task of minimizing the number of sensors required to cover all discrete targets can be equivalently mapped to the well-known *set-covering problem* in its canonical form. It is worth noting that the decision version of this problem, which determines whether a subset of $\mathcal{F}$ with a given size k can cover all the targets, has been proven to be NP-complete [76]. Consequently, it follows that the Q-Coverage problem is NP-hard, and thus the problem formulated in this thesis is also classified as NP-hard. □

3.2 Application of the problem

In practical scenarios, the reliability of sensor nodes is a concern, making it essential to monitor targets with higher precision or establish a fault-tolerant system. In such cases, simple coverage is often insufficient, leading to the adoption of a

concept known as k-coverage. k-coverage ensures that each target is monitored by at least k sensor nodes, where k is a fixed integer constant.

When a target T is covered by k sensors, the failure of up to $k-1$ nodes does not impact the continuous monitoring of T by at least one sensor. Similarly, even if there are k node-disjoint paths from a target T to the base station, the connectivity of T to the base station remains intact even if up to $k-1$ paths fail (i.e. if any node on those $k-1$ paths fails).

Moreover, employing k-coverage contributes to energy efficiency in the sensor network through effective sensor scheduling, which, in turn, extends the overall network lifetime. By activating nodes only when necessary and deactivating them during periods of inactivity, energy conservation is achieved.

However, a limitation of k-coverage is its uniform treatment of all targets, which may not be appropriate in real-world scenarios where some targets or areas demand higher priority and more extensive supervision. To address this concern, the concept of Q-coverage is introduced, where the minimum number of sensor nodes required to cover each target varies based on their respective importance. Additionally, Q-connectivity accounts for different targets needing varying numbers of node-disjoint paths to the base station for maintaining connectivity. This thesis tackles both the problems of Q-Coverage and Q-Connectivity in 3D environment, allowing it to be easily adapted to real-life use case when the surveillance area is not a simple 2D region.

3.3 Related works

The challenges of coverage and connectivity is a familiarizing concept in network deployment. They are the basis of data exchange and consequently reliable network performance. Thus, there have been various studies on this area of network deployment. This section investigates recent work on WSNs in terms of coverage and connectivity constraints.

a, Coverage

This part presents a concise overview of related works on coverage constraint in WSNs, including 1-Coverage, K-Coverage, and Q-Coverage.

1-Coverage. Arivudainambi et al. [21] presented a vertex coloring based sensor deployment (VC-SD) algorithm to determine sensor requirement and optimal spot placement and to obtain 100% target coverage in a 3D region. Mini et al. [77] treated the 3D deployment problem as a clustering problem to be solved with an artificial bee colony algorithm to determine optimal spot placement for minimum

sensing range sensors. Additionally, the variation in sensing range and whether node locations are near optimal are studied through the sensitivity analysis test. Chen et al. [78] proposed an improved ant lion optimizer for maximum network coverage by combining Cuckoo Search, Cauchy mutation and differential evolution, which is then shown to have higher convergence speed and accuracy as well as ability to skip local optima. Song et al. [79] proposed a solution using the change step of fruit fly optimization algorithm along with construction of two mathematical network coverage models. Njoya et al. [80] solved the problem of simple coverage with the objective of minimizing the number of nodes deployed using a new method involving virtual sensors. Virtual Sensor is basically a concept where a sensor can be active or inactive during the process of determining the location of nodes before the actual deployment itself. Virtual sensors can move, merge, recombine or explode, again, during the course of the algorithm, not after the deployment of nodes. Two works [12], [14] introduce an approach for 1-coverage at any point in the surveillance area by utilizing regular pattern.

K-Coverage. Kim et al. [67] proposed three regular patterns to ensure p-coverage at any point in an area of interest. A lowerbound of p is determined by the type of pattern used. Wang et al. [81] tackled K-coverage different constraints for wireless sensor networks in 3D underwater regions by a K-Equivalent Radius enhanced Virtual Force Algorithm which can satisfy diverse K-coverage requirements for practical application. Ozdag et al. [82] proposed Maximum Area Detection Algorithm based on Whale Optimization Algorithm to for the K-coverage in 2D problem, which notably outperformed previously suggested Maximum Area Detection Algorithm based on Electromagnetism-Like Algorithm in terms of energy consumption. Gupta et al. [63] inspected the problem of placing sensor nodes in pre-fixed potential positions so that k-coverage and m-connectivity are fulfilled and the number of positions selected is minimized. They proposed a metaheuristic approach based on GA. The proposed algorithm was shown to have far better time complexity and yielded better results than other GA-based approaches. Hanh et al. [23] proposed a heuristic approach to tackle K-coverage in 2D domain. Their proposal is proved to be superior to the prior GA approach.

Q-Coverage. Balaji et al. [22] proposed solving the problem obliging Q-Coverage constraint in 3D for deterministic deployment with an evolutionary algorithm called particle swarm optimization algorithm. This approach has accomplished simulation results significantly outperforming random deployment. Afizudeen et al. [83] proposed the Grundy Number based approach for the coverage and connectivity constraints of the Q-Coverage problem in 3D terrain. This method

uses Grundy Number based Deterministic Sensor Deployment and provides further improvements in optimality using Grundy Number based Random Deployment algorithm, Grundy Number based Cuckoo Search algorithm and Grundy Number based Genetic algorithm. Singh et al. [84] investigated the optimal deployment for Q-coverage through a matheuristic approach and used a genetic algorithm within a column generation framework. This approach targets basic Q-coverage problem as well as that with variable energy consumption and that with base station connectivity requirement. Hanh et al. [20] solved Q-coverage problem in 2D space by a heuristic approach based on MIP (mixed interger programming). The result is superior to other baseline methods but has high time complexity.

b, Connectivity

This part introduces our survey on existing works regarding connectivity constraint, consisting of 1-Connectivity, K-Connectivity, and Q-Connectivity. Note that $1, K, Q$ are just the notations, and different paper may use other expressions with the same concept.

1**-Connectivity.** Arivudainambi et al. [21] utilized Breadth-first search (BFS) to verify the connectivity between sensors in the network. Afzudeen et al. [83] did the same thing but proposed adding the relay nodes if the connectivity is not achieved. Both of these works operating on 3D environment.

K**-Connectivity.** Most of the multi-connectivity research studies focus on K-connectivity problem, in which each node is connected to K other nodes. These nodes can be sensors or relay nodes. One approach is using Genetic Algorithm (GA) to optimize the number of nodes [63]–[66]. The nodes are placed in predefined positions, which is encoded in the chromosomes. Some other works solve the problem of finding optimal regular pattern providing K-connectivity with some small value of K [12], [14], [67]. A sensor deployment pattern is called regular if the Voronoi polygon generated by each sensor node has the same shape and size. A different definition of K-Connectivity is presented in these works, in which there are at least K disjoint paths between any node. With that concept, Pu et al. [13], proposed a partial K-connectivity repair algorithm, in which they minimized the number of nodes to add in a network so that it becomes partial K-connected (i.e, it guarantees K disjoint paths only between special pair of nodes). Hanh et al. [23] introduced an improved Djikstra algorithm to solve K-connectivity problem in 2D area with the aim to minimize the number of relay nodes.

Q**-Connectivity.** Our survey reveals that there is only one study solving this problem. Hanh et al. [20] proposed a novel Q-Connectivity concept which is target-

Table 3.2: Comparison of related works on target coverage and connectivity problems.

Authors	***Year***	***Dimensions***		***Coverage***			***Connectivity***		
		2D	**3D**	1	K	Q	1	K	Q
Arivudainambi et al. [21]	2020		✓	✓			✓		
Mini et al. [77]	2010		✓	✓					
Chen et al. [78]	2022	✓		✓					
Song et al. [79]	2018	✓		✓					
Njoya et al. [80]	2017	✓		✓					
Bai et al. [12]	2008	✓		✓				✓	
Yun et al. [14]	2010	✓		✓				✓	
Kim et al. [67]	2012	✓			✓			✓	
Wang et al. [81]	2019		✓		✓				
Ozdag et al. [82]	2018	✓			✓				
Gupta et al. [63]	2016	✓			✓			✓	
Hanh et al. [23]	2022	✓			✓			✓	
Balaji et al. [22]	2018		✓			✓			
Afizudeen et al. [83]	2023		✓			✓	✓		
Singh et al. [84]	2013	✓				✓	✓		
Hanh et al. [20]	2023	✓				✓			✓
Gupta et al. [64]	2016	✓			✓			✓	
Mini et al. [65]	2012	✓						✓	
Srivastava et al. [66]	2019	✓						✓	
Pu et al. [13]	2009	✓						✓	

centric. Each target t_i has q_i node-disjoint paths to the base station through the corresponding sensors covering it. The authors used a maximum flow approach combined with clustering algorithm to minimize the number of relay nodes.

The conclusion drawn is that though there are numerous solutions developed by scholars to deploys sensor nodes in diverse-scaled regions, most have been under ideal predefined environment behavior, thus making realistic adaptation challenging. Specifically, much of the research has been conducted on the 2D plane or only tackled 1-coverage and K-coverage whilst many real-world scenarios requires more complex 3D deployment to which 2D techniques cannot be extended to. Moreover, none of the related works solve the Q-Connectivity in 3D environment. Table 3.2 shows the summary of comparisons of the target coverage and connectivity problem with different initial constraints.

CHAPTER 4. PROPOSED METHODS

This chapter encompasses a comprehensive overview of the two-phase framework devised to address the problem introduced in the preceding chapter. Additionally, it presents detailed algorithms for each phase, namely SPARTA for phase I, and GEMSTONE for phase II.

4.1 Two-phase framework

To handle the formulated problem, this thesis proposes a two-phase graph-based approach. In the first phase, the sensor locations are optimized to satisfy Q-Coverage. Then, in the second phase, the relay node coordinates are optimized to satisfy Q-Connectivity. These two phases are solved independently with two different algorithms. The flow chart of the approach is presented in Figure 4.1. The problem solved in each phase is elaborated below.

Phase I

Phase I addresses the Q-Coverage problem, which can be defined as follows:

Input:

- Target set $T = \{t_1, t_2, \ldots, t_n\}$,
- Priority vector $Q = \{q_1, q_2, \ldots, q_n\}$,
- Sensing range r_s.

Output: Sensor set $S = \{s_1, s_2, \ldots, s_m\}$

Objective: Minimize $|S|$

Constraint: Ensure that each target t_i is simultaneously covered by q_i sensors, for all $i = 1, 2, \ldots, n$ (referred to as Q-Coverage).

The Q-Coverage problem involves taking the target set, priority vector, and sensing range as input and producing the sensor set as output. The objective is to minimize the number of sensors while satisfying the Q-Coverage constraint.

Phase II

Phase II addresses the Q-Connectivity problem, which can be defined as follows:

Input:

- Target set $T = \{t_1, t_2, \ldots, t_n\}$,
- Priority vector $Q = \{q_1, q_2, \ldots, q_n\}$,

- Communication range r_c,
- Sensor set $S = \{s_1, s_2, \ldots, s_m\}$,
- Base station location B.

Output: Relay node set $R = \{r_1, r_2, \ldots, r_p\}$

Objective: Minimize $|R|$

Constraint: Ensure that each target t_i has q_i node-disjoint paths to the base station B through q_i sensors covering it (referred to as Q-Connectivity).

The Q-Connectivity problem involves taking the target set, priority vector, communication range, sensor set obtained from Phase I, and base station location as input, and producing the relay node set as output. The objective is to minimize the number of relay nodes while satisfying the Q-Connectivity constraint.

In the following section, the algorithms for each phase are introduced, namely SPARTA for phase I and GEMSTONE for phase II.

4.2 SPARTA: Sphere Point And Removal Target Algorithm

4.2.1 Vanilla SPARTA

In phase I, we propose a algorithm with the main idea of gradually placing sensors into good overlap regions and remove satisfied targets until there are no targets left.

Firstly, we find the good overlap regions through examining the intersection points of spheres in the sphere set O. In this thesis, to determine such points, we take 3 spheres at once and calculate the intersection of them. This is also known as the famous Trilateration problem in geopositioning. The solution can be easily calculated [85]. Accordingly, there can be $0, 1$ or 2 intersection points between 3 spheres (see Figure 4.2). Such points constitute the set I^{triad}.

In order to take advantage of more overlap regions, we also calculate the midpoint of each target pair if their two corresponding spheres intersect (see Figure 4.3). Those points constitute the set I^{pair}. Then we can obtain the set $I = I^{triad} + I^{pair}$.

Note that with each point p found during the above process, p^{cover}, p^{parent} are computed. Besides, t^{child} and t^{sensor} are also calculated for each target t.

After constructing the intersection set I, we perform the following loop until there are no targets left:

- Select 2 points I_i and I_j in I which cover the same and largest set of targets.

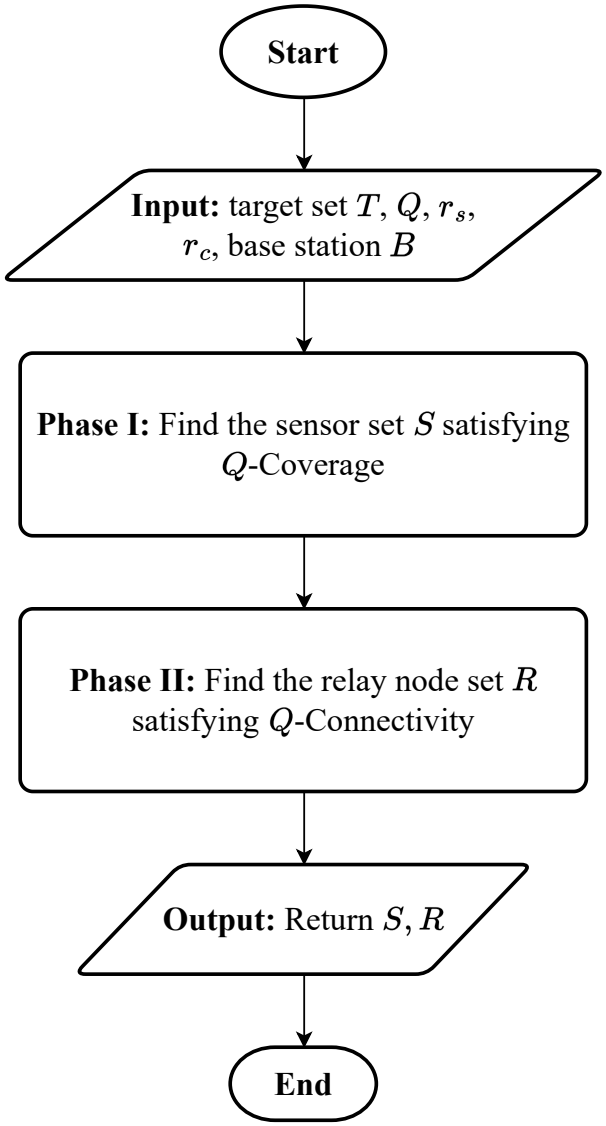
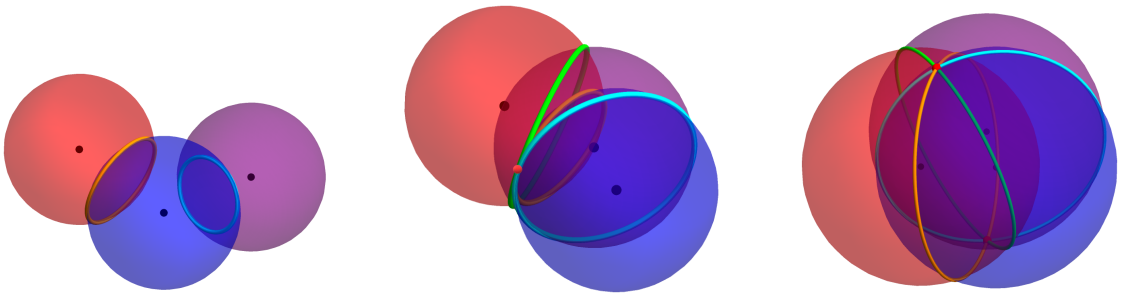


Figure 4.1: Flow chart of the two-phase approach.



(a) No intersection. **(b)** 1 intersection. **(c)** 2 intersections.

Figure 4.2: Possible number of intersection points of 3 spheres. Intersection points are marked in red.

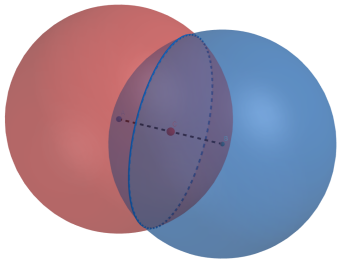


Figure 4.3: Illustration of mid point of two spheres (marked in red).

- Place minimum sensor between I_i and I_j so that at least the coverage constraint of one target in I_i^{cover} is satisfied.
- Remove the targets whose coverage constraint is satisfied and update the coverage constraint of other targets in I_i^{cover}.
- If $I = \emptyset$, exit the loop. At this point, the remaining targets do not have any intersections with each other. Place the remaining necessary number of sensors inside each corresponding sphere and end the algorithm.

Figure 4.4 illustrates a simple case with 5 targets. The pseudo code for SPARTA is given in Algorithm 5.

Theorem 2. *Assuming that $n >> q_{max}$, the time complexity of SPARTA is $O(n^5)$.*

Proof. Firstly, finding the set $I = I^{triad} + I^{pair}$ takes $O(n^3)$ since we have to loop through all triads and pairs, of which the quantities are $O(n^3)$ and $O(n^2)$, respectively.

Afterwards, we perform iterations until $T = \emptyset$. In the worst case, we need to loop n times. Within each loop, we first update p^{cover} for all $p \in I$. This takes $O(n^4)$ as we have to loop through all remaining targets (which is $O(n)$) and all remaining elements in I (which is $O(n^3)$), and each append operation takes only $O(1)$.

Next, we consider the worst case where $I \neq \emptyset$. In this case, we need to sort the set I, which contains the intersection of triplets and pairs of spheres. The cardinality of I is $O(n^3)$. Therefore, the sorting step takes $O(n^3 \log n)$.

The step of finding $q_{min} = \min_{t \in I_i^{cover}} q_t$ takes $O(n)$ since I_i^{cover} has $O(n)$ elements.

The placement of q_{min} sensors takes $O(q_{max})$.

The loop over the elements in I_i^{cover} takes about $O(n(n^3 + n))$ because I_i^{cover} has $O(n)$ elements, and the removal of t takes $O(n)$ time, while the removal of t_{child} takes approximately $O(n^3)$ time.

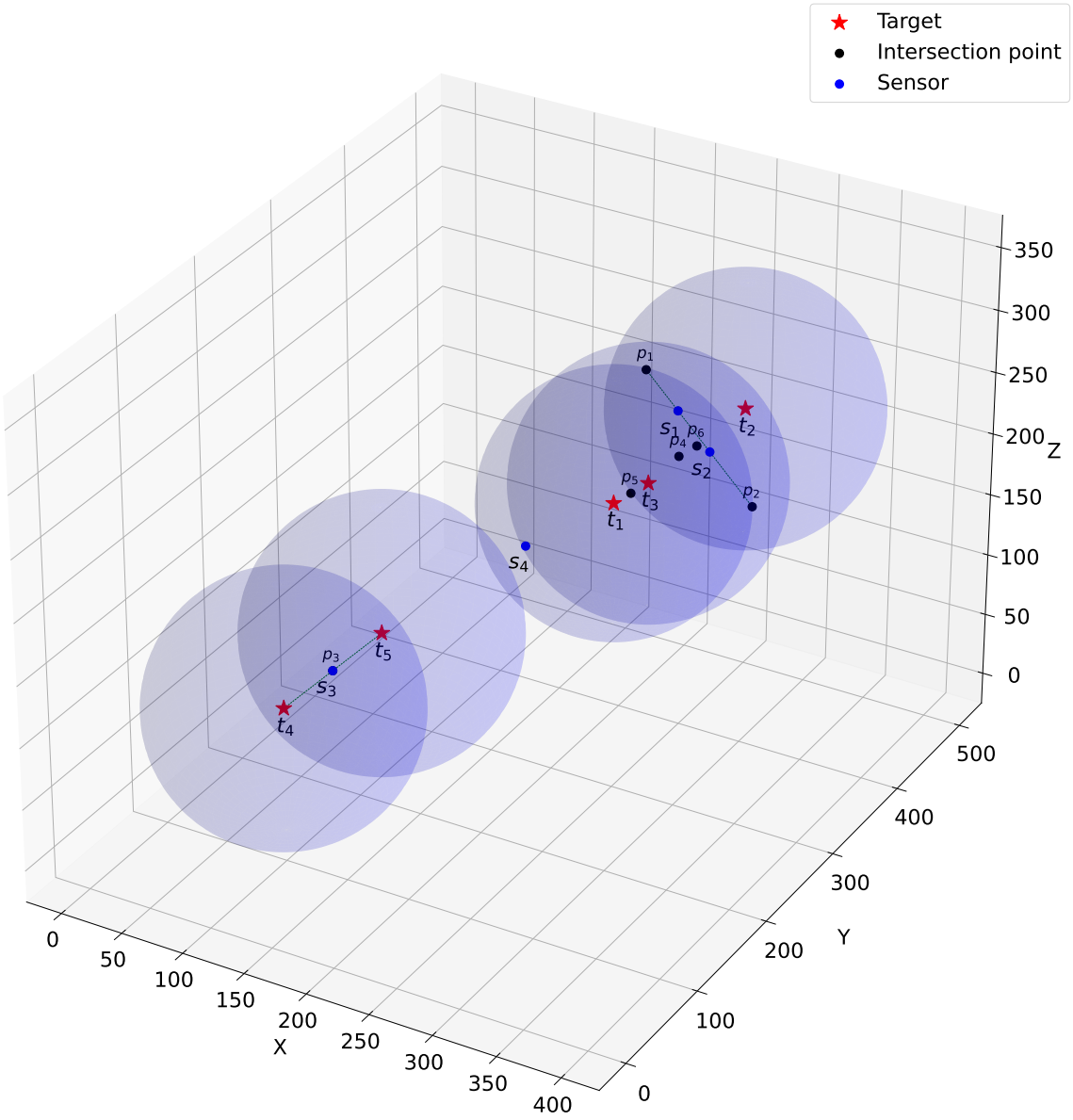


Figure 4.4: Illustration of the result of SPARTA in a case with 5 targets $\{t_1, t_2, t_3, t_4, t_5\}$, given $q_1 = 3, q_2 = q_3 = 2, q_4 = q_5 = 1$. The set $I^{triad} = \{p_1, p_2\}$ and $I^{pair} = \{p_3, p_4, p_5, p_6\}$.

Overall, the total process takes $O(n^3 + n(n^4 + n^3 \log n + q_{max} + n + n(n^3 + n))) = O(n^3 + n(n^4 + q_{max})) = O(n^5)$ as $n >> q_{max}$. □

Note that the above theorem only provides a loose upper bound of the time complexity of SPARTA. In fact, we can efficiently reduce this upper bound by a simple technique. Suppose that we can divide the target set into k subsets and process each set independently. If the initial T has n elements then the complexity of the algorithm is $O(n^5)$. Hence if T is divided into k subsets, each has approximately $\frac{n}{k}$ elements, then the total complexity for executing SPARTA in k subsets is $k * O((\frac{n}{k})^5) = O(\frac{n^5}{k^4})$, which means the upper bound of complexity is reduced roughly k^4 times.

Hence we introduce two innovative strategies for partitioning the target set, which are clique partitioning (CP) and connected component (CC). In both strategies, a graph $G(V, E)$ is constructed with V is the target set. There is an edge between 2 vertexes if and only if the Euclidean distance between them is not greater than $2r_s$ (i.e., two corresponding spheres intersect). The details are presented in the following sections.

4.2.2 SPARTA-CC: SPARTA with Connected Component

In SPARTA-CC, we find all the connected components in the graph with DFS. Finally, SPARTA is executed on each component. Since there are no intersections between two different components, SPARTA-CC gives the same result as SPARTA but in significantly less time. According to the preliminaries of this thesis, the time complexity for finding all connected components in a graph $G(V, E)$ is $O(|V| + |E|)$. In this case, $|V| = n$. Thus the time complexity of SPARTA-CC is $O(n + |E|) + O(n^5/k_{cc}^4) = O(n + |E| + n^5/k_{cc}^4)$ with k_{cc} being the number of connected components and $|E|$ being the number of edges in the graph.

4.2.3 SPARTA-CP: SPARTA with Clique Partition

In the case that the network is dense enough, almost every target belongs to the same connected component, thus SPARTA-CC does not differ much from vanilla SPARTA. Therefore, we leverage the maximal clique partition [75] to split the target set. A clique is a subgraph G' of G so that G' is a complete graph. This strategy can still preserve the intersective relation between spheres but produce smaller thus more partitions compared to SPARTA-CC. This can lead to a reduction in computational time. According to the preliminaries of this thesis, the time complexity for finding the clique-partition in a graph $G(V, E)$ is $O(|V|^2)$. In this case, $|V| = n$. Thus the time complexity of SPARTA-CC is $O(n^2) + O(n^5/k_{cc}^4) = O(n^2 + n^5/k_{cp}^4)$ with k_{cp} being the number of cliques.

In Phase II, the sensor set S obtained in Phase I is used as input to determine the locations of relay nodes that satisfy Q-Connectivity. To achieve this, we introduce an algorithm in the following section that is based on the concept of minimum spanning tree (MST).

4.3 GEMSTONE: Graph-based Efficient Minimum Spanning Tree Optimization for Network Establishment

The main idea of GEMSTONE is partitioning the sensor set S from Phase I into k disjoint sensor subsets $S_1, S_2, \ldots, S_k$ so that with each target t, all sensors in t^{sensor} belong to different subsets (*). Then with each subset S_i, we can define a complete graph $G_i(V_i, E_i)$ with $V_i = S_i + B$. In each graph, a minimum spanning tree can be formed, thus providing an unique path from each sensor to B. Accordingly, a target t which is covered by a set t^{sensor} of q_t sensors will have q_t node-disjoint connections to B, each corresponding to a different sensor subset.

With the above description, it can be deduced that the number of subsets k must not less than q_{max}. For the sake of simplicity, we choose $k = q_{max}$. The process of creating vertex sets $V_i (i \in \{1, 2, \ldots, k\})$ is illustrated in Figure 4.5, 4.6 and Algorithm 6. Specifically, the details of it are elaborated as follows:

- In the first step, each column j corresponds to the set t_j^{sensor}.
- Then in column j, the sensors are reordered so that if a sensor s appears in column $j' < j$, it must appear in the same row as it does in column j' (see Figure 4.5).
- In each row, duplicated sensors are removed. Finally, add B to row i to form the set V_i.

It can be proved that the subsets created by the above process satisfy the condition (*). After finding the MST for each graph G_i, in the final step, relay nodes are placed along the tree edges. The whole process of the two-phase algorithm is illustrated in Figure 4.7, 4.8, and 4.9.

Theorem 3. *Denote l_{max} as the maximum length of an edge in any MST found by GEMSTONE. Assuming that $n >> l_{max}/r_c$, the time complexity of GEMSTONE is $O(q_{max}^2 n^2)$.*

Proof. Firstly, the sorting step takes $O(n \log n)$.

After that, we have three nested loops. Considering that t_k^{sensor} has $O(q_{max})$ elements and finding an element A in it takes $O(q_{max})$, the total time complexity of these loops is $O(n^2 q_{max}^2)$.

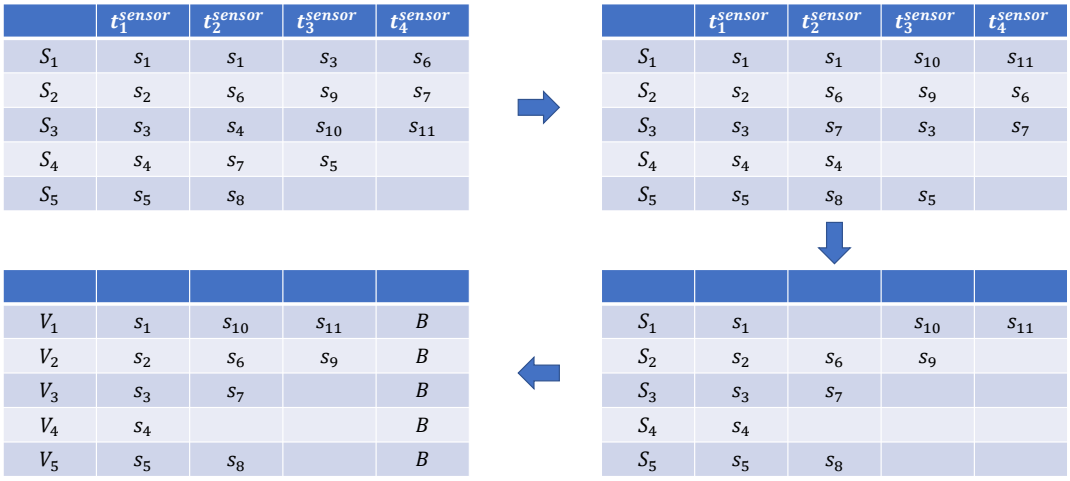


Figure 4.5: Illustration of creating vertex sets $V_i (i \in \{1, 2, 3, 4, 5\})$. There are four targets with $q_{max} = 5$.

The next loop to form the set V_i takes $O(nq_{max})$.

Ultimately, since Prim's algorithm implemented with adjacency list and Fibonacci heap takes $O(E + V \log V)$ with E and V being the number of edges and vertexes in a graph, respectively. As $E = O(n^2)$ and $V = O(n)$ in our case, the time complexity of Prim's algorithm is $O(n^2)$. Placing relay nodes in $O(n)$ edges of the graph takes $O(nl_{max}/r_c)$, since there are $O(l_{max}/r_c)$ relay nodes on each edge. Hence, the complexity of the final loop is $O(q_{max}(n^2 + nl_{max}/r_c)) = O(q_{max}n^2)$.

Therefore, the time complexity of GEMSTONE is $O(n \log n + n^2 q_{max}^2 + nq_{max} + q_{max}n^2) = O(q_{max}^2 n^2)$. This represents a tight upper bound for the time complexity of GEMSTONE.

□

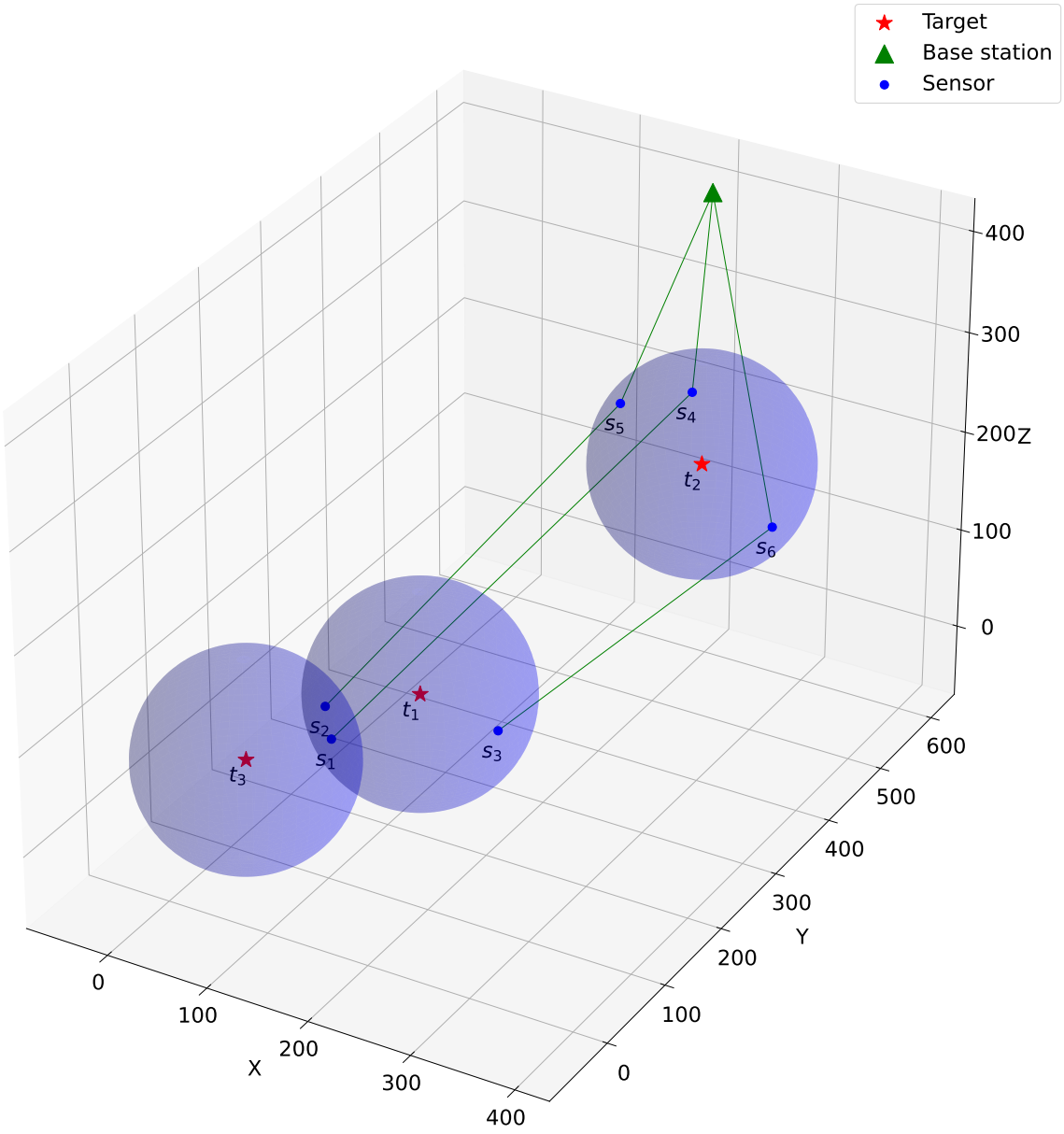


Figure 4.6: Illustration of creating MST in each vertex set $V_i (i \in \{1, 2, 3\})$. There are three targets with $q_{max} = 3$. In this case, $V_1 = \{s_1, s_4, B\}, V_2 = \{s_2, s_5, B\}, V_3 = \{s_3, s_6, B\}$.

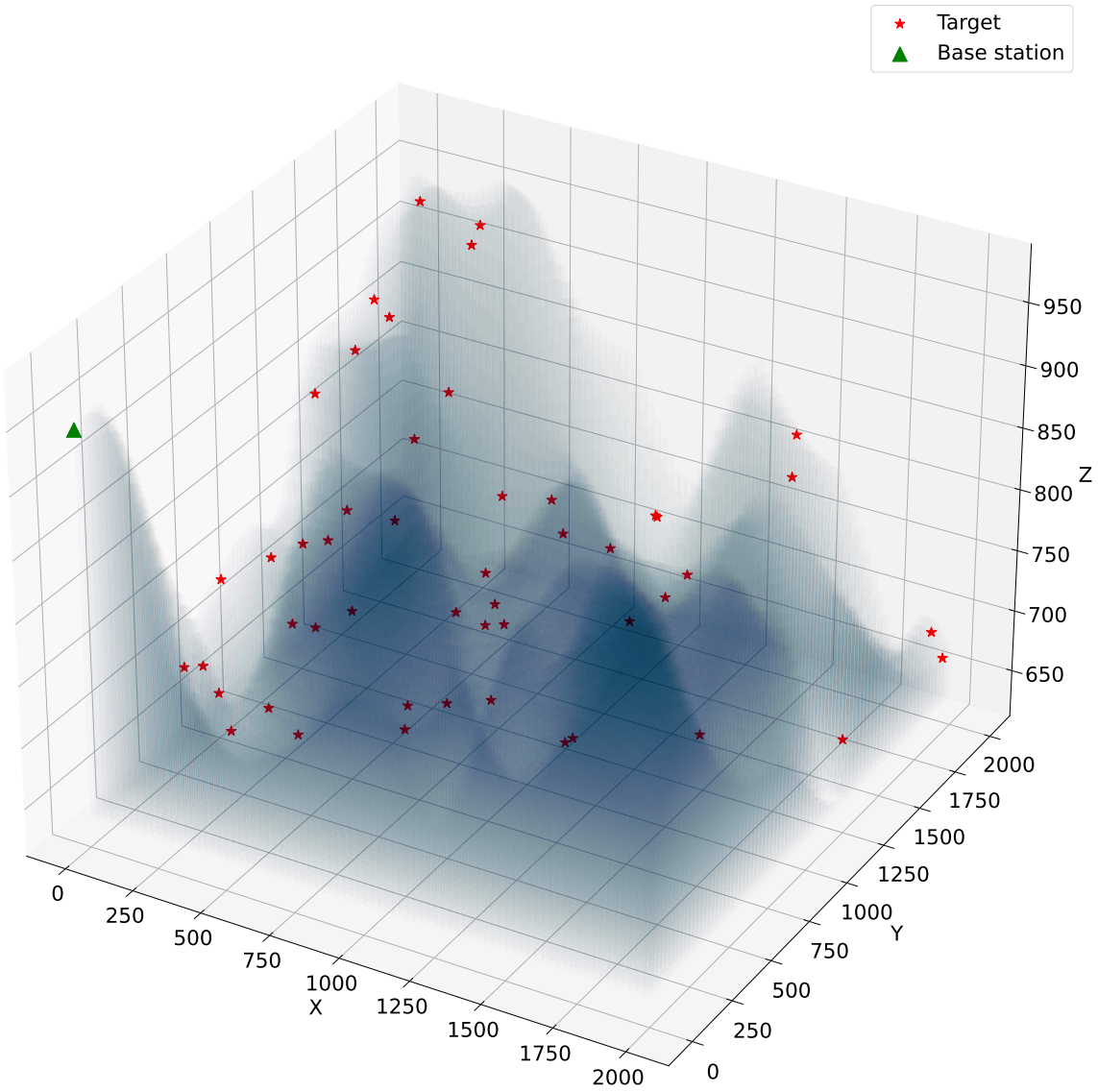


Figure 4.7: Illustration of 50 targets in 3D environment. The $x-$ and $y-$coordinates are generated randomly in a 2000×2000 area. Suppose that the height at any given coordinate (x, y) is known.

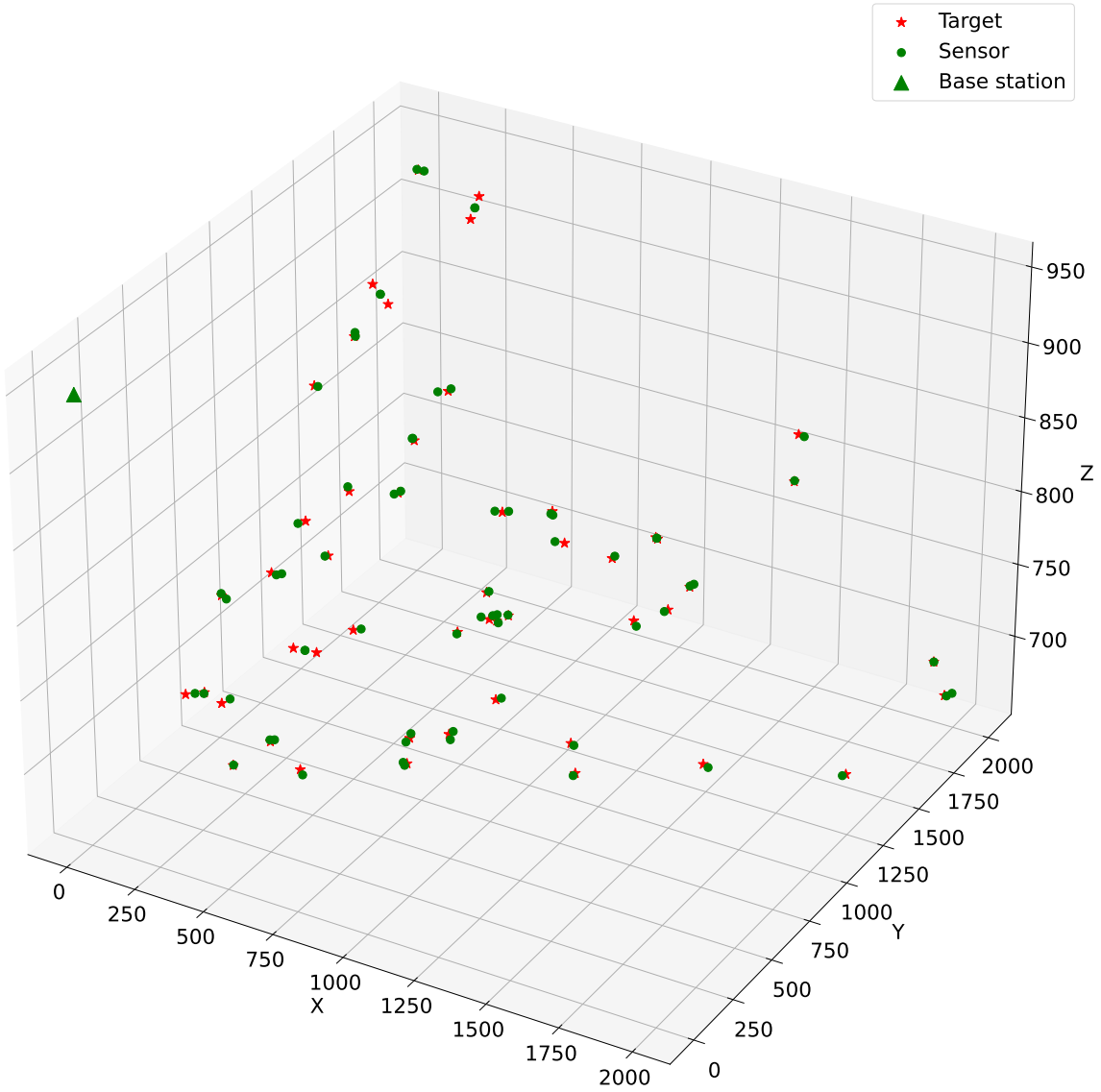


Figure 4.8: Result illustration of phase I when $r_s = 40\ (m)$ and $q_{max} = 2$. Sensors are placed to satisfy Q-Coverage.

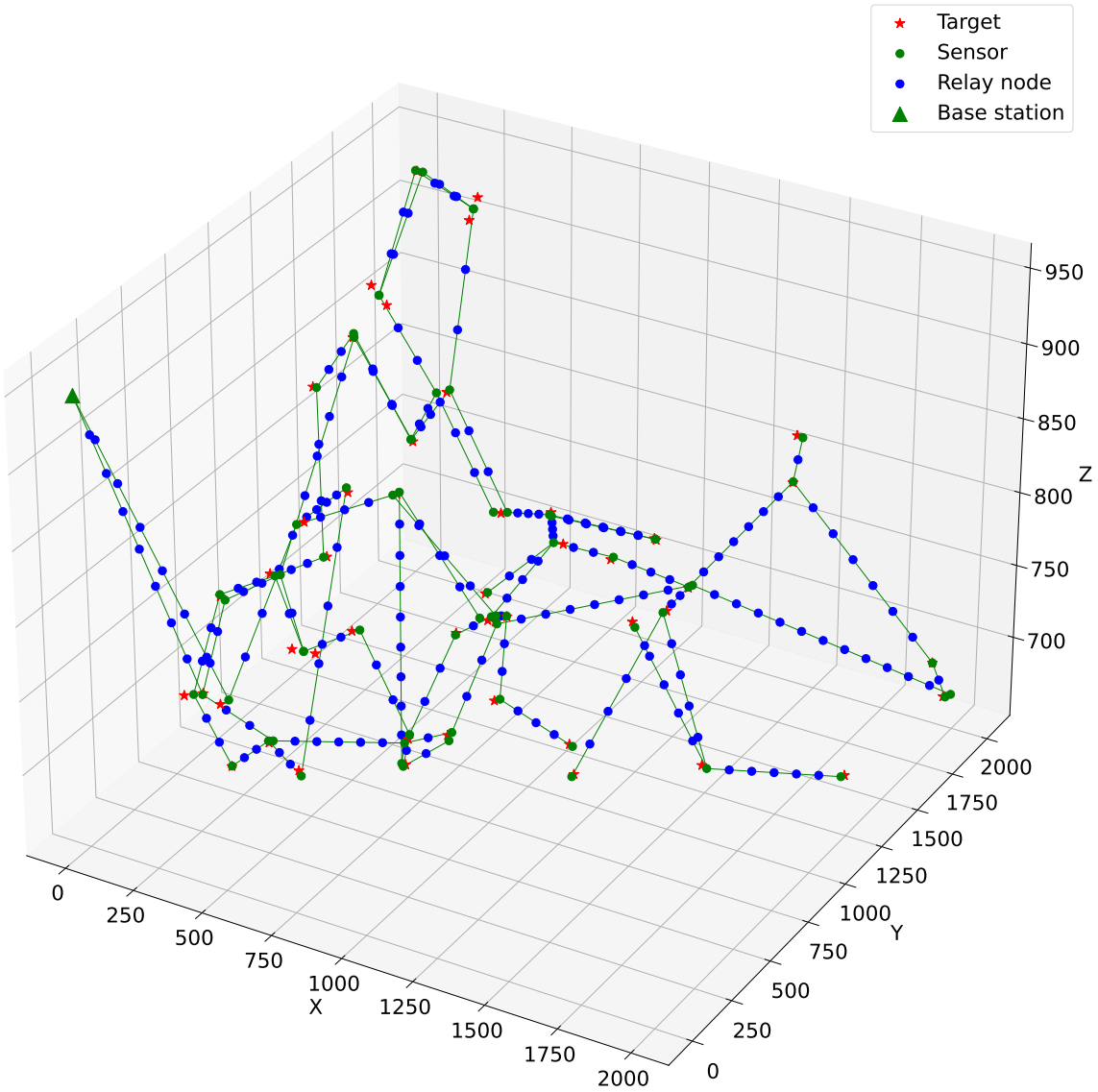


Figure 4.9: Result illustration of phase II when $r_c = 80\ (m)$ and $q_{max} = 2$. Relay nodes are placed to satisfy Q-Connectivity. The connections between them and the base station are also plotted.

Algorithm 5: SPARTA: **S**phere **P**oint **A**nd **R**emoval **T**arget **A**lgorithm

Input : n: number of targets
$T = \{t_1, t_2, \ldots, t_n\}$: list of targets
Vector Q: q_i: priority level of target $i, i = 1..n$
r_s: sensing range
$A = L \times W \times H$: surveillance region

Output: Location of the minimum number of sensors needed to satisfy Q-Coverage $S = [(x_1, y_1, z_1), (x_2, y_2, z_2), ..., (x_m, y_m, z_m)]$.

Build sphere set O.
$I =$ **FindIntersection**(O)
while $T \neq \emptyset$ **do**
    Update p^{cover} for all $p \in I$.
    if $I \neq \emptyset$ **then**
        Select 2 point I_i, I_j in I which have the largest $|I_l^{cover}|$ and $I_i^{cover} = I_j^{cover}$.
        $q_{min} = \min_{t \in I_i^{cover}} q_t$
        Place q_{min} sensors S_{tmp} between I_i and I_j.
        for t *in* I_i^{cover} **do**
            $t^{sensor} += S_{tmp}$
            $q_t -= q_{min}$
            if $q_t = 0$ **then**
                Remove t.
                Remove t^{child} from I.
            end
        end
    end
    else
        for $t \in T$ **do**
            Place q_t random sensors S_{tmp} in O_t.
            $t^{sensor} += S_{tmp}$
            Remove t from T.
        end
    end
end

Algorithm 6: GEMSTONE: **G**raph-based **E**fficient **M**inimum **S**panning **T**ree **O**ptimization for **N**etwork **E**stablishment

Input : n: number of targets
$T = \{t_1, t_2, \ldots, t_n\}$: list of targets
Vector Q: q_i: priority level of target $i, i = 1..n$
r_c: communication range
$A = L \times W \times H$: surveillance region
B: base station

Output: Location of the minimum number of relay nodes needed to satisfy Q-Connectivity $R = [(x_1, y_1, z_1), (x_2, y_2, z_2), ..., (x_u, y_u, z_u)]$.

Sort t in descending order of t^{sensor} length.
for $i = 2$ *to* n **do**
  for $A \in t_i^{sensor}$ **do**
    for $k = 1$ *to* $i - 1$ **do**
      if $A \in t_k^{sensor}$ **then**
        $B = t_k^{sensor}$.index(A)
        t_i^{sensor}.swap(A, B)
      end
    end
  end
end
$V = \emptyset$
for $i = 1$ *to* q_{max} **do**
  for $j = 1$ *to* n **do**
    $V_i += \{t_j^{sensor}{}_i\}$
  end
  $V_i =$ set(V_i)
  $V_i += \{B\}$
end
for $i = 1$ *to* q_{max} **do**
  $E_i =$ **MinimumSpanningTree**(V_i)
  Place relay nodes along the edges in E_i.
end

CHAPTER 5. NUMERICAL RESULTS

This chapter provides a detailed account of the experiments conducted as part of the thesis. It begins by introducing the dataset employed in the experiments. Next, the parameter settings utilized in the experiments are presented. Subsequently, the various experiment scenarios are outlined. Finally, the numerical results obtained from the experiments are presented, accompanied by an in-depth analysis of the findings.

5.1 Dataset

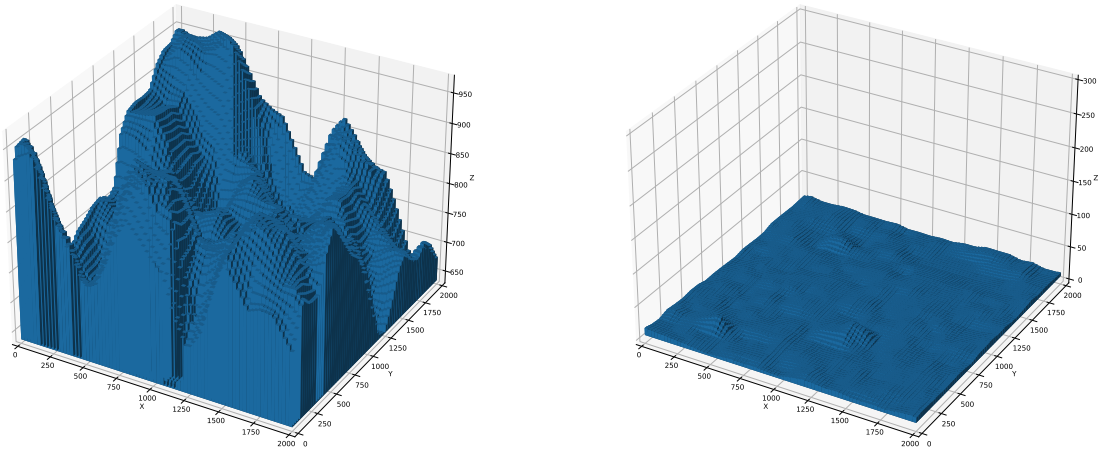
Due to the absence of an accessible public dataset pertaining to the given problem, this thesis develops two new distinct datasets derived from divergent geographical settings, namely AoI 1 and AoI 2. AoI 1 has high variation in altitude. Meanwhile, AoI 2 has low variation in altitude. Each dataset corresponds to a region measuring $2000 \times 2000(m^2)$. Both datasets comprise an altitude matrix of dimensions 80×80, where each element denotes the elevation of a $25 \times 25(m^2)$ square area. The illustrations of these datasets are shown in Figure 5.1, where each entry in the matrix is plotted as a bar with the corresponding height. The altitude matrix creates a surface on which the target locations are later generated randomly on. Note that in this thesis, obstacles in the environment are not considered. One example is shown in Figure 4.7.

5.2 Parameter settings

The proposals are implemented in Python 3.9 and executed on AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz, RAM 16GB DDR4 3200MHz.

Besides, the default parameters and dataset used in the experiments are presented as follows. Note that the default values of r_s, r_c and q_{max} are adopted from [20].

- The default dataset is AoI 1 because AoI 2 does not have much difference with a 2D dataset (i.e., the variation in height is not significant).
- $n = 400$. n 3D target coordinates are randomly generated on the surface of the dataset.
- $r_s = 40(m)$
- $r_c = 80(m)$
- $q_{max} = 10$. The priority value of each target is randomly generated in the range $[1, q_{max}]$.



(a) AoI 1 (high variation in altitude).

(b) AoI 2 (low variation in altitude).

Figure 5.1: Illustrations of two datasets.

5.3 Experiment scenarios

In each phase, we conduct comprehensive experiment scenarios to assess the impact of each factor to the performance of the proposed algorithms, as well as compare them to the baselines. The details of the scenarios are elaborated below.

5.3.1 Q-Coverage

In phase I, we compare SPARTA, SPARTA-CC and SPARTA-CP with two other baselines. We keep the default setting and implementation in the original papers and run all the algorithms on our dataset for a fair comparison. The details of the baselines are introduced below.

- **GLA** [20] is a method utilizing the Mixed Integer Programming (MIP) to determine the number of sensors to place in the optimal regions found by the greedy algorithm. Since the original GLA aims to solve Q-Coverage on 2D environment, we apply the same process in SPARTA to find the optimal regions of GLA so that it can work on 3D environment.
- **GN-DSD** [83] applies the Vertex Coloring Algorithm to partition the target set into multiple groups, each of which can be covered by only one sensor. The algorithm also gradually remove satisfied targets and ultimately attempt to remove redundant sensors.

There are three scenarios assessing the impact of changing n, r_s, q_{max} and environment type, as follows:

- **Scenario 1.1.** This scenario evaluates the impact of changing n to the algorithm performance. The number of targets varies from 100 to 850 with stride

Table 5.1: Scenario 1.1 parameters.

n	$r_s(m)$	q_{max}	**Dataset**
100	40	10	AoI 1
250			
400			
550			
700			
850			

Table 5.2: Scenario 1.2 parameters.

n	$r_s(m)$	q_{max}	**Dataset**
400	20	10	AoI 1
	35		
	50		
	65		
	80		
	95		

of 150. There are six network instances involved, which are presented in Table 5.1.

- **Scenario 1.2.** In this particular scenario, we assess the effect of modifying the sensing range (r_s) on the performance of the algorithm. The sensing range is adjusted in increments of $15(m)$, ranging from $20(m)$ to $95(m)$. This evaluation involves six different network instances, which are outlined in Table 5.2.
- **Scenario 1.3.** This scenario focuses on assessing the influence of the maximum priority value (q_{max}) on the algorithm's performance. We vary q_{max} in increments of 2, ranging from 2 to 10. The evaluation is conducted using five different network instances, as detailed in Table 5.3.
- **Scenario 1.4.** In this particular scenario, we examine the influence of the environment on the performance of the algorithm. The experiments are conducted using two datasets while keeping other parameters fixed as default. This

Table 5.3: Scenario 1.3 parameters.

n	$r_s(m)$	q_{max}	**Dataset**
400	40	2	AoI 1
		4	
		6	
		8	
		10	

Table 5.4: Scenario 1.4 parameters.

n	$r_s(m)$	q_{max}	**Dataset**
400	40	10	AoI 1
			AoI 2

evaluation encompasses two network instances, which are detailed in Table 5.4.

For each scenario and network instance, we calculate the average number of sensors and runtime as the performance metrics. To obtain reliable results, we repeat the experiments 20 times for each algorithm and record the average values.

5.3.2 Q-Connectivity

In the second phase of our method, we conduct a comparative analysis between our proposed GEMSTONE algorithm and two other algorithms as introduced below:

- **CMFA** [20] constructs target clusters and establishes intraconnection within each cluster, as well as interconnection among clusters. To ensure a fair comparison, we adopt the best configuration settings as reported in the original paper, while maintaining the unchanged implementation for CMFA since it can be directly applied to 3D area.
- **FCSA**, as proposed in [20], is a straightforward method wherein every target generates q_i connections between itself and the base station using its corresponding set of sensors. Similar to CMFA, FCSA can be seamlessly implemented in a 3D setting without requiring any adjustments or modifications.

The input to all algorithms in phase II is the sensor locations produced by SPARTA-CC in phase I. Besides, the base station is located at coordinate $(0, 0, z)$, where z is the height of point $(0, 0)$ in the selected dataset. To evaluate the impact of different parameters and environmental conditions, we design four distinct scenarios, namely:

- **Scenario 2.1.** This scenario investigates the influence of varying the number of targets (n) on the algorithm's performance. We consider a range of target values from 100 to 850 with a stride of 150. A total of six network instances are utilized for evaluation, as presented in Table 5.5.
- **Scenario 2.2.** In this scenario, we examine the effect of adjusting the sensing range (r_c) on the algorithm's performance. The sensing range is incrementally modified in steps of $15(m)$, ranging from $80(m)$ to $155(m)$. This evaluation

Table 5.5: Scenario 2.1 parameters.

n	$r_s(m)$	$r_c(m)$	q_{max}	**Dataset**
100	40	80	10	AoI 1
250				
400				
550				
700				
850				

Table 5.6: Scenario 2.2 parameters.

n	$r_s(m)$	$r_c(m)$	q_{max}	**Dataset**
400	40	80	10	AoI 1
		95		
		110		
		125		
		140		
		155		

encompasses six distinct network instances, as outlined in Table 5.6.

- **Scenario 2.3.** In this scenario, our aim is to evaluate the impact of the maximum priority value (q_{max}) on the algorithm's performance. We systematically vary q_{max} in increments of 2, spanning from a minimum value of 2 to a maximum value of 10. To conduct this evaluation, we utilize five distinct network instances, which are comprehensively described in Table 5.7.
- **Scenario 2.4.** In this scenario, we investigate the impact of the environment on the algorithm's performance. The experiments are conducted using two distinct datasets while keeping other parameters at their default values. This evaluation includes two network instances, which are described in Table 5.8.

For each scenario and network instance, we calculate the average number of relay nodes and runtime as performance metrics. These metrics are obtained by running the algorithms 20 times and averaging the results.

Table 5.7: Scenario 2.3 parameters.

n	$r_s(m)$	$r_c(m)$	q_{max}	**Dataset**
400	40	80	2	AoI 1
			4	
			6	
			8	
			10	

Table 5.8: Scenario 2.4 parameters.

n	$r_s(m)$	$r_c(m)$	q_{max}	**Dataset**
400	40	80	10	AoI 1
				AoI 2

5.3.3 Combined two phases

In this experimental scenario, with the aim of assessing the performance of our comprehensive two-phase algorithm, SPARTA-GEMSTONE, alongside two of its variants is evaluated. We conduct a comparative evaluation with the two-phase approach proposed by Hanh et al. [20], denoted as GLA-CMFA. All experiments are conducted using the default parameter values mentioned in Section 5.2, ensuring consistency and fairness in the evaluation process.

To obtain statistically significant results, each algorithm is executed 20 times, and the resulting metrics are averaged for further analysis. The performance metrics considered for comparison encompass the average number of nodes and average runtime in each phase, as well as the total number of nodes and total runtime.

5.4 Experiment results

5.4.1 Q-Coverage results

- **Scenario 1.1.**

  Figure 5.2 and Table 5.9 depict the outcomes of Scenario 1.1, which focuses on the evaluation of the number of sensors and computational time. In terms of sensor count, SPARTA-CC outperforms all other algorithms and achieves the same result as vanilla SPARTA (as proved in the proposal), while GN-DSD exhibits the least favorable performance. GLA emerges as the second-best algorithm, surpassing SPARTA-CP by a marginal improvement. Specifically, SPARTA-CC is 1.04 times better than GLA and 1.39 times better than GN-DSD when $n = 850$. The superiority of SPARTA-CC over SPARTA-CP can be attributed to its ability to effectively exploit the overlapping regions among target spheres. Conversely, GN-DSD suffers from two major drawbacks that contribute to its inferior performance: its reliance on cliques, which prevents it from capitalizing on the overlap area between cliques, and its inefficient strategy for eliminating redundant sensors due to sequential erasing without comprehensive consideration of the erasing order's impact.

  Regarding runtime, GLA exhibits the slowest performance among the four algorithms due to its time-consuming mixed integer programming approach. SPARTA has nearly the same running time as GLA. Besides, SPARTA-CC

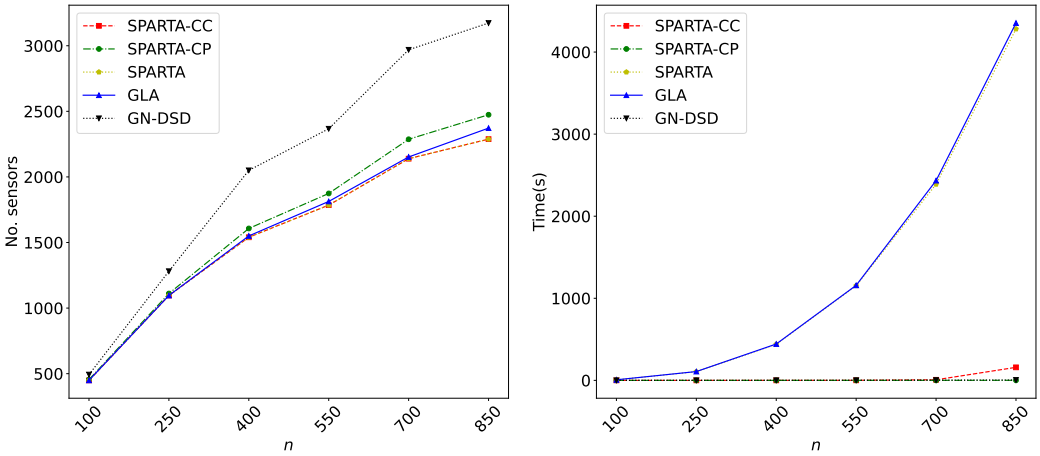


Figure 5.2: Scenario 1.1 results.

Table 5.9: Scenario 1.1 results.

	n	*100*	*250*	*400*	*550*	*700*	*850*
No. sensors	GN-DSD	493	1281	2050	2367	2969	3173
	GLA	449	1096	1550	1813	2152	2372
	SPARTA-CP	455	1111	1607	1874	2287	2474
	SPATAR-CC	**449**	**1096**	**1540**	**1785**	**2139**	**2288**
	SPARTA	**449**	**1096**	**1540**	**1785**	**2139**	**2288**
Time(s)	GN-DSD	0.02254	0.19131	0.62371	1.25556	2.55208	4.21693
	GLA	6.73569	107.25488	442.89113	1157.84351	2434.01801	4353.35800
	SPARTA-CP	0.00561	0.04163	0.12429	**0.26488**	**0.47871**	**0.74173**
	SPATAR-CC	**0.00239**	**0.01186**	**0.10326**	0.68261	6.92089	158.63480
	SPARTA	6.60351	106.06493	439.47009	1153.20148	2389.50916	4280.96800

proves to be the fastest when the number of targets (n) is small ($n \leq 400$), while SPARTA-CP showcases the fastest performance in all other cases. Particularly, SPARTA-CC can reduce the running time of vanilla SPARTA by 4256 times in the case $n = 400$. When n is small, the disparities in cluster number and cardinality between SPARTA-CC and SPARTA-CP are negligible. Moreover, the time required for finding connected components is shorter than the time needed for identifying the maximal clique partition. Consequently, SPARTA-CC demonstrates slightly superior runtime. However, as n increases, the cluster size in SPARTA-CC escalates rapidly, causing SPARTA-CC to become slower than SPARTA-CP.

- **Scenario 1.2.**

  Figure 5.3 and Table 5.10 provide the results of Scenario 1.2, where the impact of increasing the sensing range (r_s) is examined. Regarding the number of sensors, SPARTA-CC achieves superior performance in all the cases. GLA remains the second-best algorithm, surpassing SPARTA-CP by a small margin. GN-DSD exhibits the least favorable performance among the four algorithms.

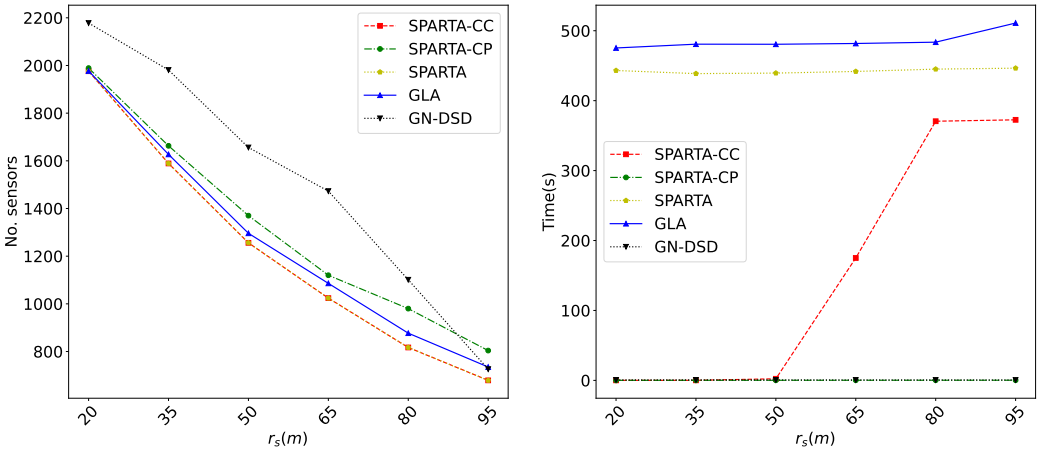


Figure 5.3: Scenario 1.2 results.

Table 5.10: Scenario 1.2 results.

	$r_s(m)$	*20*	*35*	*50*	*65*	*80*	*95*
No. sensors	GN-DSD	2178	1981	1656	1473	1101	726
	GLA	1977	1627	1296	1086	877	735
	SPARTA-CP	1989	1663	1370	1120	980	804
	SPATAR-CC	**1976**	**1589**	**1256**	**1024**	**817**	**679**
	SPARTA	**1976**	**1589**	**1256**	**1024**	**817**	**679**
Time(s)	GN-DSD	0.63355	0.63226	0.60424	0.56801	0.53902	0.50817
	GLA	475.35775	480.76933	480.64530	481.75905	483.63470	510.98250
	SPARTA-CP	0.06972	0.11798	**0.14490**	**0.16947**	**0.19943**	**0.22170**
	SPARTA-CC	**0.02225**	**0.05478**	2.29128	175.03751	370.58600	372.59840
	SPARTA	443.12558	438.68054	439.42220	441.77010	445.09630	446.46020

In a particular case $r_s = 65(m)$, SPARTA-CC is 1.06 times better than GLA and 1.44 times better than GN-DSD. In all cases, the number of sensors decreases as r_s increases due to the increased overlap between target spheres, resulting in more optimal regions for sensor placement.

Figure 5.3 also reveals that the runtime of most algorithms experiences only a slight increase as r_s increases, except for SPARTA-CC. Nevertheless, it is still 1.37 times faster than GLA and 1.20 times faster than vanilla SPARTA when $r_s = 95(m)$. This can be attributed to the rapid growth of the size of connected components in SPARTA-CC when r_s increases, whereas cliques in SPARTA-CP face greater difficulty in expanding their size. GLA remains the slowest algorithm among the four, while SPARTA-CP exhibits the fastest runtime when r_s is large (i.e., more overlapping). When $r_s = 95(m)$, it is 2305 times faster than GLA and 2.29 times faster than GN-DSD. The same reasons mentioned in Scenario 1.1 contribute to this outcome.

- **Scenario 1.3.**

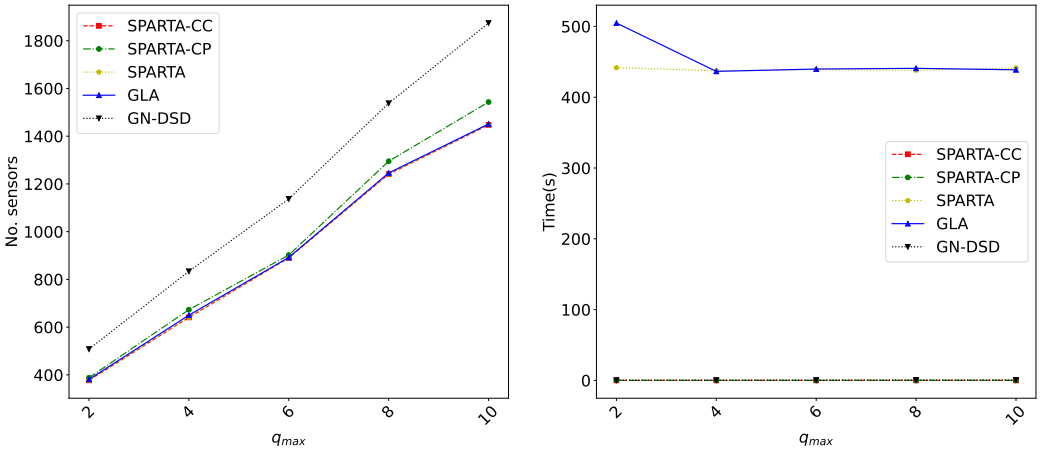


Figure 5.4: Scenario 1.3 results.

The outcomes of Scenario 1.3 are presented in Figure 5.4 and Table 5.11. The results of all algorithms exhibit nearly linear scaling in relation to the variable q_{max}, which represents the maximum allowable quantity of sensors. This behavior can be attributed to the fact that the number of sensors deployed during each iteration of the SPARTA-family algorithm scales linearly with respect to q_{max}. Among the algorithms considered, SPARTA-CC and SPARTA consistently yield the best performance, followed by GLA and SPARTA-CP as the second and third best, respectively, with only marginal differences in their results. Conversely, GN-DSD remains the least effective algorithm in this particular scenario. Specifically, GN-DSD is 1.3 times worse than SPARTA-CC when $q_{max} = 10$.

Regarding runtime, SPARTA-CC proves to be the fastest algorithm across all values of q_{max} examined. The subsequent ordering in terms of increasing runtime is SPARTA-CP, GN-DSD, SPARTA and GLA. The runtime of all algorithms exhibits minimal changes as q_{max} increases due to the fact that the number of iterations remains approximately constant, with only the number of sensors placed during each iteration increasing linearly. Given that the values of n and r_s are fixed, SPARTA-CC consistently achieves the best runtime in all cases where the level of overlap is not substantial enough, as elucidated in previous scenarios. It is 4392 times faster than vanilla SPARTA, 4367 times faster than GLA and 5.78 times faster than GN-DSD when $q_{max} = 10$.

- **Scenario 1.4.**

The findings from Scenario 1.4, investigating the influence of environment type, are depicted in Figure 5.5 and Table 5.12. The outcomes reveal that the

Table 5.11: Scenario 1.3 results.

q_{max}		*2*	*4*	*6*	*8*	*10*
No. sensors	GN-DSD	508	834	1137	1538	1874
	GLA	381	650	892	1246	1451
	SPARTA-CP	388	673	902	1295	1543
	SPATAR-CC	**377**	**641**	**890**	**1241**	**1447**
	SPARTA	**377**	**641**	**890**	**1241**	**1447**
Time(s)	GN-DSD	0.20780	0.29585	0.38291	0.52870	0.58103
	GLA	504.73460	436.49640	439.79020	440.79210	438.67990
	SPARTA-CP	0.12060	0.12012	0.12269	0.13145	0.13275
	SPARTA-CC	**0.09725**	**0.09876**	**0.10312**	**0.10046**	**0.10046**
	SPARTA	441.68790	437.32890	438.82250	438.17060	441.26360

Table 5.12: Scenario 1.4 results.

Dataset		AoI 2	AoI 1
No. sensors	GN-DSD	1979	2050
	GLA	1500	1550
	SPARTA-CP	1549	1607
	SPARTA-CC	**1486**	**1540**
	SPATAR	**1486**	**1540**
Time(s)	GN-DSD	0.61159	0.62371
	GLA	452.38630	442.89110
	SPARTA-CP	**0.12793**	0.12429
	SPARTA-CC	0.13573	**0.10326**
	SPARTA	460.41956	439.47009

uneven-height area (AoI 1) necessitates a greater number of sensors compared to the even-height area (AoI 2), despite both scenarios having the same sensing range (r_s) and q_{max}. Although the value of n remains constant across the two environments, the high variation in altitude of AoI 1 poses challenges for achieving target sphere overlap. Consequently, the average Euclidean distance between targets becomes longer, leading to a slight increase in the required sensor quantity.

Regarding runtime, the reduced overlap in the flatter environment of AoI 2 results in decreased number of elements in the set I, which leads to less time for sorting, updating the cover set for each point, etc. Moreover, in AoI 2, cluster sizes of both SPARTA-CC and SPARTA-CP also decreases. Additionally, the number of variables in the Mixed Integer Programming (MIP) formulation of GLA also decreases. These modifications contribute to reduced runtime for SPARTA and GLA when operating in a flatter environment. Conversely, in the case of GN-DSD, as it places sensors one by one, its runtime increases with the growing number of required sensors.

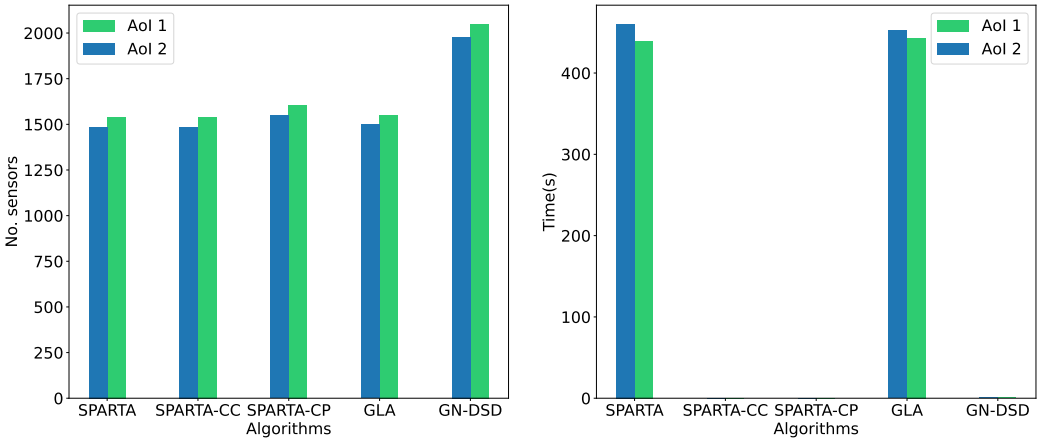


Figure 5.5: Scenario 1.4 results.

5.4.2 Q-Connectivity results

- **Scenario 2.1.**

  Table 5.13 and Figure 5.6 present the findings from Scenario 2.1, which examines the influence of the target count. Among the various algorithms, GEMSTONE emerges as the most favorable in terms of the number of relay nodes. Specifically, when the number of targets is set to $n = 850$, GEMSTONE achieves a result that is 2.02 times superior to CMFA and 21.81 times better than the FCSA algorithm, which follows a naive approach.

  Our analysis reveals that regarding the number of relay nodes, CMFA performs poorly compared to our proposed algorithm for two main reasons. Firstly, CMFA's simplistic mechanism of repeatedly selecting the shortest remaining edge and evaluating conditions exhibits a greedy nature that hinders its efficiency. Secondly, CMFA relies on a strong assumption that the paths derived from a node in the graph formed by cluster centers must be node-disjoint to fulfill the Q-connectivity constraint. However, this assumption is unnecessary and ultimately leads to inferior outcomes.

  As for runtime considerations, GEMSTONE significantly outperforms CMFA, being 129.31 times faster when $n = 850$, while also achieving superior relay node counts. This discrepancy can be attributed to CMFA's utilization of maximum flow algorithms, which are employed multiple times corresponding to the number of edges in the final graph. This iterative process proves time-consuming. In contrast, GEMSTONE utilizes the Prim's algorithm (which is faster than maximum flow algorithms) just one round for every vertex set. Besides, GEMSTONE's runtime remains comparable to that of FCSA, which fol-

lows a straightforward and naive approach. The difference in runtime between GEMSTONE and FCSA is negligible.

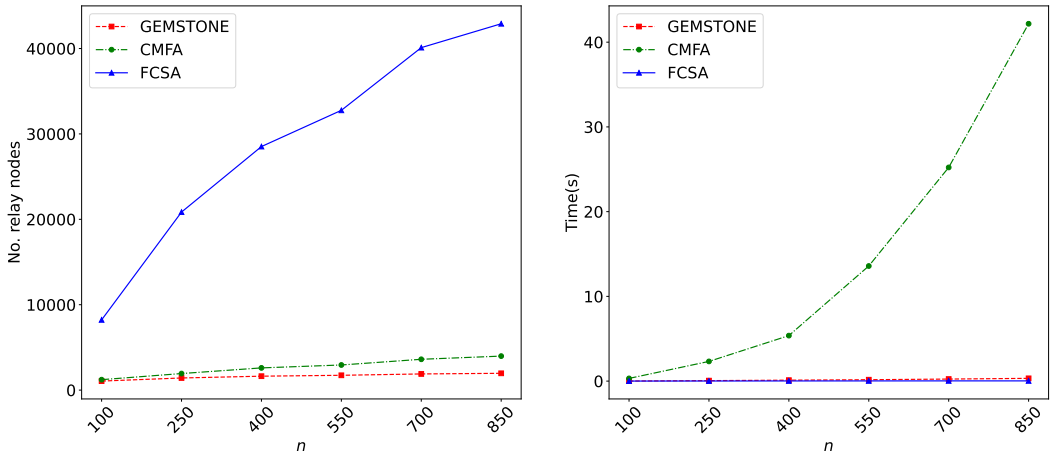


Figure 5.6: Scenario 2.1 results.

Table 5.13: Scenario 2.1 results.

n		*100*	*250*	*400*	*550*	*700*	*850*
No. relay nodes	FCSA	8224	20847	28519	32762	40104	42900
	CMFA	1223	1939	2601	2942	3609	3982
	GEMSTONE	**1058**	**1420**	**1630**	**1734**	**1893**	**1967**
Time(s)	FCSA	**0.00444**	**0.01270**	**0.01886**	**0.02269**	**0.02694**	**0.02578**
	CMFA	0.31839	2.32429	5.36840	13.58329	25.21656	42.17305
	GEMSTONE	0.01064	0.05641	0.11085	0.16450	0.24051	0.32613

- **Scenario 2.2.**

  The outcomes of Scenario 2.2, investigating the influence of the communication range r_c, are depicted in Figure 5.7 and Table 5.14. It is important to note that the value of r_c does not impact the number of links formed by the algorithms but only affects the placement of relay nodes on each link. Consequently, the number of relay nodes is inversely proportional to r_c, resulting in hyperbolic graph shapes.

  With regard to the number of relay nodes, GEMSTONE continues to outperform all other algorithms across all scenarios. Specifically, when $r_c = 155$ meters, GEMSTONE exhibits a superiority of 3.17 times over CMFA and 48.68 times over FCSA.

  Regarding computational time, the duration required for placing relay nodes is significantly lower compared to that for identifying connections, particularly when the number of targets is substantial. As the connections remain unchanged for different values of r_c, the total runtime of all algorithms remains

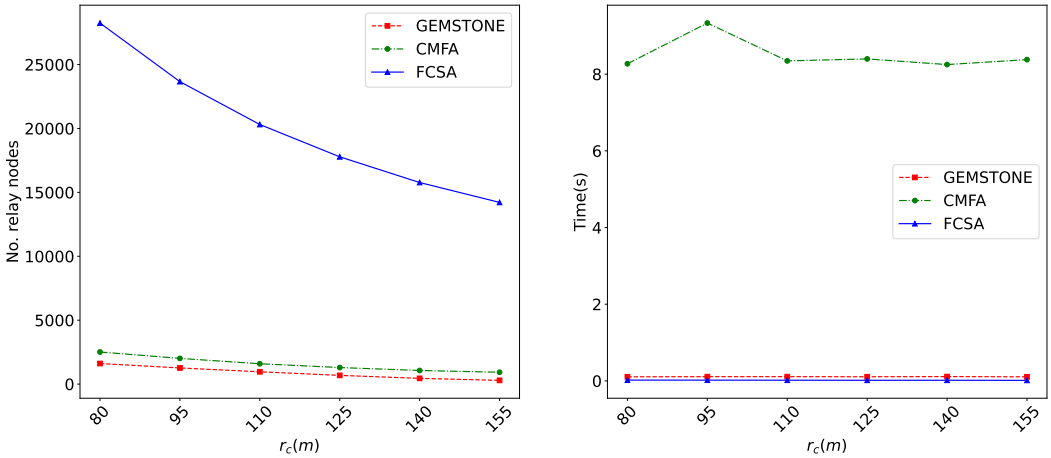


Figure 5.7: Scenario 2.2 results.

Table 5.14: Scenario 2.2 results.

$r_c(m)$		*80*	*95*	*110*	*125*	*140*	*155*
No. sensors	FCSA	28247	23663	20309	17782	15771	14214
	CMFA	2511	2012	1592	1295	1064	927
	GEMSTONE	**1610**	**1262**	**964**	**682**	**449**	**292**
Time(s)	FCSA	**0.02099**	**0.01928**	**0.01790**	**0.01597**	**0.01647**	**0.01381**
	CMFA	8.27101	9.33300	8.34619	8.39751	8.25033	8.37900
	GEMSTONE	0.10620	0.11004	0.11018	0.10758	0.11217	0.10618

approximately constant. Furthermore, GEMSTONE proves to be approximately 80 times faster than CMFA and achieves a comparable speed to that of FCSA in this scenario.

- **Scenario 2.3.**

  The results from Scenario 2.3, examining the impact of the variable q_{max}, are presented in Table 5.15 and Figure 5.8. In terms of the number of relay nodes, GEMSTONE continues to outperform the other algorithms. Specifically, when $q_{max} = 10$, GEMSTONE exhibits a superiority of 1.55 times over CMFA and 17.10 times over FCSA. Additionally, it can be observed that the number of relay nodes for all algorithms increases linearly in relation to q_{max}.

  In regards to runtime, GEMSTONE is approximately 79 times faster than CMFA when $q_{max} = 10$ while still achieving a comparable speed to FCSA. The Ford-Fulkerson Algorithm employed by CMFA possesses a time complexity of $O(\text{maxflow} \times E)$, where E represents the number of edges in the graph. In our specific problem, the maximum flow cannot exceed q_{max}. Therefore, the computational time of CMFA scales linearly with q_{max}.

- **Scenario 2.4.**

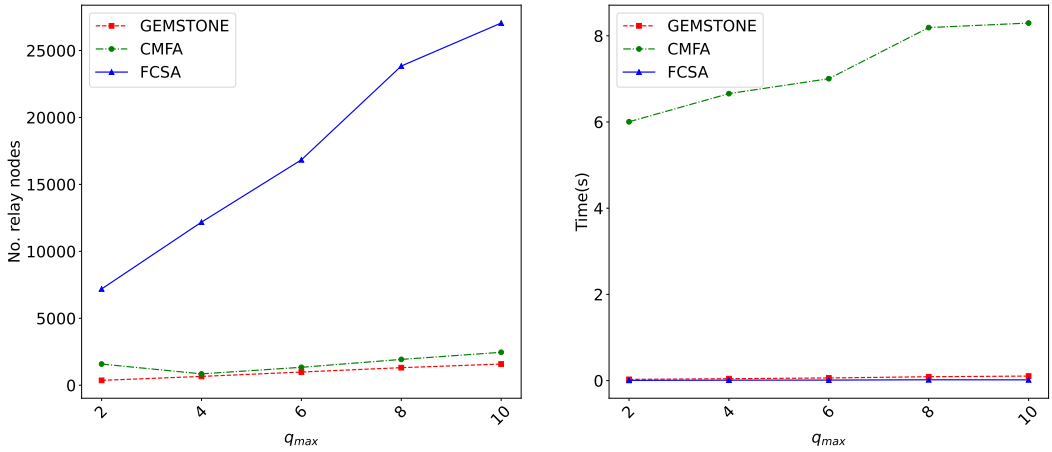


Figure 5.8: Scenario 2.3 results.

Table 5.15: Scenario 2.3 results.

	q_{max}	*2*	*4*	*6*	*8*	*10*
No. sensors	FCSA	7189	12181	16834	23842	27051
	CMFA	1581	848	1338	1928	2458
	GEMSTONE	**363**	**657**	**985**	**1311**	**1582**
Time(s)	FCSA	**0.00521**	**0.00981**	**0.01183**	**0.01854**	**0.01699**
	CMFA	6.00477	6.65820	7.00477	8.18999	8.29570
	GEMSTONE	0.02769	0.04374	0.06098	0.08943	0.10461

The findings from Scenario 2.4, investigating the influence of different environment types, are presented in Table 5.16 and Figure 5.9. It is observed that all algorithms yield a slightly higher number of relay nodes when operating in AoI 1, which is a uneven-height area. This can be attributed to the larger distances between targets in such terrain, necessitating the placement of more nodes to establish connections between the targets and the base station. This observation aligns with the explanation provided in Scenario 1.4.

Table 5.16: Scenario 2.4 results.

Dataset		AoI 2	AoI 1
No. sensors	FCSA	27270	28519
	CMFA	2524	2601
	GEMSTONE	**1527**	**1630**
Time(s)	FCSA	**0.01663**	**0.01886**
	CMFA	4.87683	5.3684
	GEMSTONE	0.10594	0.11085

In terms of computational time, all algorithms demonstrate slightly faster runtimes in the flat area of AoI 2. This disparity arises from the fact that in the

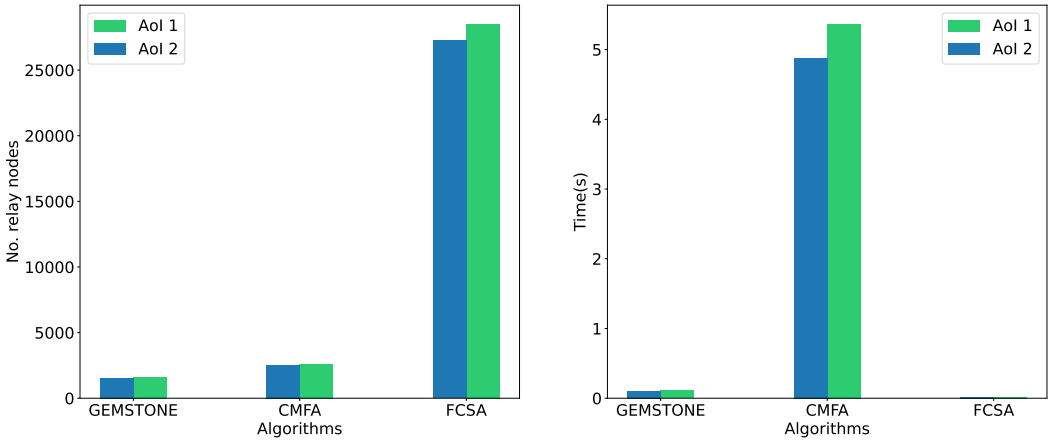


Figure 5.9: Scenario 2.4 results.

uneven-height area of AoI 1, there are more sensors and less overlap between target spheres, which leads to more vertexes in each vertex subset V_i overall. Hence, the MST finding algorithm takes longer to run. Moreover, there is a greater number of relay nodes to be positioned in AoI 1, resulting in a longer time required for node placement compared to AoI 2. Additionally, for CMFA, the sparser distribution of targets in AoI 1 leads to an increase in the number of clusters (i.e., the number of vertices in the graph), which further contributes to longer runtimes.

5.4.3 Combined two phases

Table 5.17: Combined two phases result.

	No. sensors	*Runtime phase I*	*No. relay nodes*	*Runtime phase II*	*No. total nodes*	*Total runtime*
GLA-CMFA	1550	442.89113	2099	64.59364	3649	507.48477
SPARTA-GEMSTONE	**1540**	439.47009	**1630**	**0.11085**	**3170**	439.58094
SPARTA-CC-GEMSTONE	**1540**	**0.10326**	**1630**	**0.11085**	**3170**	**0.21411**
SPARTA-CP-GEMSTONE	1607	0.12429	1593	0.13772	3200	0.26201

Table 5.17, Figure 5.10, and Figure 5.11 depict the outcomes obtained from the two-phase algorithms. Notably, SPARTA-CC-GEMSTONE exhibits superior performance across all evaluated metrics, significantly surpassing GLA-CMFA, particularly in terms of runtime. Precisely, SPARTA-CC-GEMSTONE achieves a speedup of 4289 times during phase I and 583 times during phase II when compared to GLA-CMFA. Overall, SPARTA-CC-GEMSTONE demonstrates a total speedup of 2370 times over GLA-CMFA. Additionally, concerning the total number of nodes, both SPARTA-CC-GEMSTONE and SPARTA-CP-GEMSTONE outperform GLA-CMFA by 1.15 and 1.14 times, respectively.

A noteworthy observation is that SPARTA-CC-GEMSTONE employs fewer

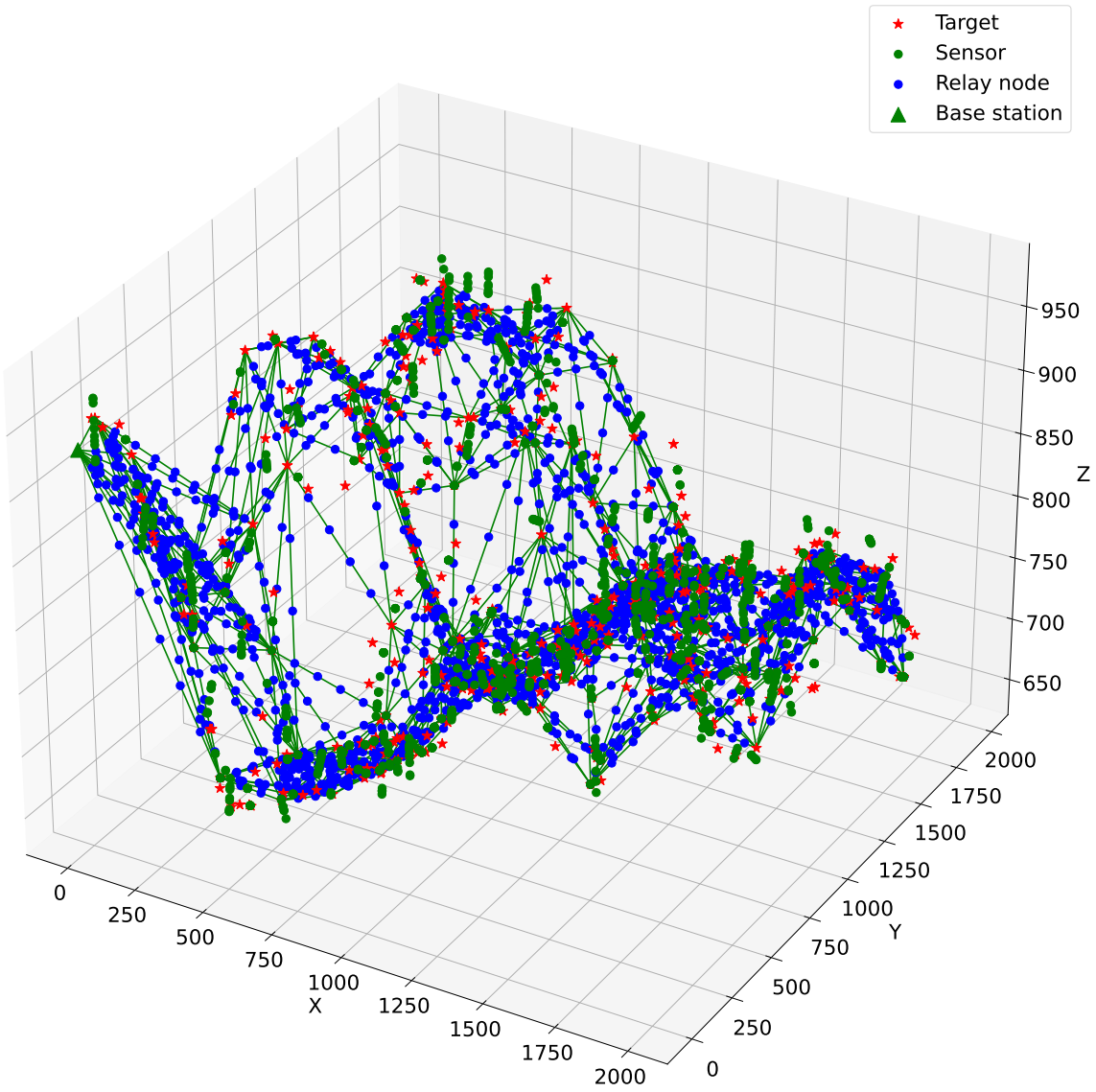


Figure 5.10: Result of GLA-CMFA with the default dataset and experiment parameters.

sensors than SPARTA-CP-GEMSTONE but relies on a higher number of relay nodes. This phenomenon indicates a trade-off between the number of sensors and relay nodes. In general, an abundance of sensors results in shorter average distances between them, leading to a reduced need for relay nodes.

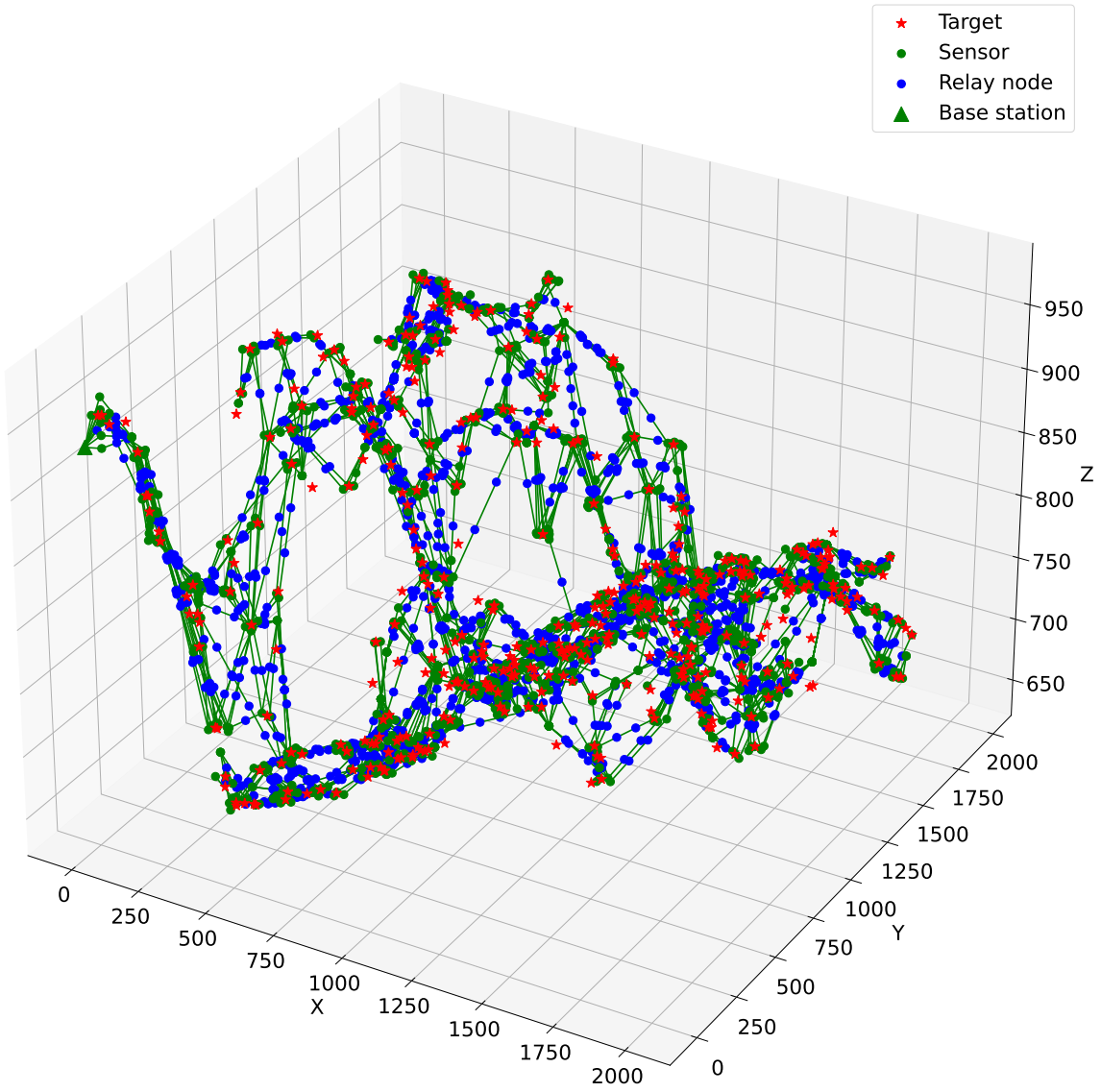


Figure 5.11: Result of SPARTA-CC-GEMSTONE with the default dataset and experiment parameters.

CONCLUSIONS

Summary

This thesis introduces the formulation of the Q-Coverage and Q-Connectivity problems and extends them to the three-dimensional space, aiming to minimize the number of nodes required. Additionally, a two-phase approach is presented to solve these problems. The first phase employs the SPARTA algorithm with a Q-Coverage constraint, while the second phase utilizes the GEMSTONE algorithm with a Q-Connectivity constraint. Comparative analysis against the state-of-the-art in both phases demonstrates that the proposed algorithms significantly outperform existing methods. In-depth analysis is provided to explain the superior performance of SPARTA and GEMSTONE. These algorithms have practical applications in constructing high-quality fault-tolerant WSNs.

A part of this research is published in "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* - ISI Q1, p. 103 578, 2023."

Suggestion for Future Works

In future research, it is possible to integrate additional practical constraints into the algorithms, such as incorporating mobile sensors, considering energy consumption, and accounting for obstacles in the environment. Furthermore, there is potential for optimizing the algorithms in terms of running time and node usage, aiming to provide even better results. These advancements would contribute to the enhanced applicability and efficiency of the proposed algorithms for addressing the given problem.

REFERENCE

[1] A. Tripathi, H. P. Gupta, T. Dutta, R. Mishra, K. K. Shukla and S. Jit, "Coverage and connectivity in wsns: A survey, research issues and challenges," *IEEE Access*, vol. 6, pp. 26 971–26 992, 2018.

[2] M. Farsi, M. A. Elhosseini, M. Badawy, H. Arafat Ali and H. Zain Eldin, "Deployment techniques in wireless sensor networks, coverage and connectivity: A survey," *IEEE Access*, vol. 7, pp. 28 940–28 954, 2019.

[3] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues and Y. Park, "Intrusion detection protocols in wireless sensor networks integrated to internet of things deployment: Survey and future challenges," *IEEE Access*, vol. 8, pp. 3343–3363, 2020.

[4] D. Kandris, C. Nakas, D. Vomvas and G. Koulouras, "Applications of wireless sensor networks: An up-to-date survey," *Applied System Innovation*, vol. 3, no. 1, 2020, ISSN: 2571-5577.

[5] J. Amutha, S. Sharma and J. Nagar, "Wsn strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues," *Wireless Personal Communications*, vol. 111, no. 2, pp. 1089–1115, 2020, ISSN: 1572-834X.

[6] B. Wang, *Coverage Control in Sensor Networks*. Springer London, 2010.

[7] S. Harizan and P. Kuila, "Nature-inspired algorithms for k-coverage and m-connectivity problems in wireless sensor networks," in *Design Frameworks for Wireless Networks*, S. K. Das, S. Samanta, N. Dey and R. Kumar, Eds. Singapore: Springer Singapore, 2020, pp. 281–301, ISBN: 978-981-13-9574-1.

[8] A. Boukerche and P. Sun, "Connectivity and coverage based protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 80, pp. 54–69, 2018, ISSN: 1570-8705.

[9] M. Karatas, N. Razi and H. Tozan, "A comparison of p-median and maximal coverage location models with q–coverage requirement," *Procedia Engineering*, vol. 149, pp. 169–176, 2016, International Conference on Manufacturing Engineering and Materials, ICMEM 2016, 6-10 June 2016, Nový Smokovec, Slovakia, ISSN: 1877-7058.

[10] Manju, P. Bhambu and S. Kumar, "Target k-coverage problem in wireless sensor networks," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 23, no. 2, pp. 651–659, 2020.

[11] P. Szczytowski, A. Khelil and N. Suri, "Dkm: Distributed k-connectivity maintenance in wireless sensor networks," in *2012 9th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*, 2012, pp. 83–90.

[12] X. Bai, D. Xuan, Z. Yun, T. H. Lai and W. Jia, "Complete optimal deployment patterns for full-coverage and k-connectivity (k≤6) wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '08, Hong Kong, Hong Kong, China: Association for Computing Machinery, 2008, 401–410, ISBN: 9781605580739.

[13] J. Pu, Z. Xiong and X. Lu, "Fault-tolerant deployment with k-connectivity and partial k-connectivity in sensor networks," *Wireless Communications and Mobile Computing*, vol. 9, pp. 909 –919, Jul. 2009.

[14] Z. Yun, X. Bai, D. Xuan, T. H. Lai and W. Jia, "Optimal deployment patterns for full coverage and k-connectivity $(k \leq 6)$ wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 934–947, 2010.

[15] R. Yarinezhad and S. N. Hashemi, "A sensor deployment approach for target coverage problem in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, 2020, ISSN: 1868-5145.

[16] S. Peng and Y. Xiong, "A lifetime-enhancing method for directional sensor networks with a new hybrid energy-consumption pattern in q-coverage scenarios," *Energies*, vol. 13, no. 4, 2020, ISSN: 1996-1073.

[17] R. Ozdag, "Optimization of target q-coverage problem for qos requirement in wireless sensor networks," *Journal of Computers*, pp. 480–489, Jan. 2018.

[18] S. Mini, S. K. Udgata and S. L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 3, pp. 636–644, 2014.

[19] D Arivudainambi, S Balaji, R Pavithra and R. Shakthivel, "Energy efficient sensor scheduling for q-coverage problem," in *2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, IEEE, 2017, pp. 1–6.

[20] 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*, p. 103 578, 2023.

[21] D. Arivudainambi and R. Pavithra, "Coverage and connectivity-based 3d wireless sensor deployment optimization," *Wirel. Pers. Commun.*, vol. 112, no. 2, 1185–1204, 2020, ISSN: 0929-6212.

[22] S. Balaji, Priyanka and Anitha, "Optimal deployement of sensors in 3d - terrain with q-coverage constraints," in *2018 IEEE SENSORS*, 2018, pp. 1–4.

[23] N. T. Hanh, H. T. T. Binh, N. Q. Huy, N. Van Thanh and B. T. Q. Mai, "Node placement optimization under k-coverage and k-connectivity constraints in wireless sensor networks," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022, pp. 01–08.

[24] N. T. T. Nguyễn Đức Nghĩa, *Toán Rời Rạc*. Nhà Xuất Bản Đại Học Quốc Gia Hà Nội, 2005, vol. 5.

[25] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[26] A. Souza, "Combinatorial algorithms," in *Lecture Notes in Computer Science*, 2019.

[27] J. S. Wilson, *Sensor technology handbook*. Elsevier, 2004.

[28] J. Brignell and N. White, *Intelligent Sensor Systems,* Taylor & Francis, 1996, ISBN: 9780750303897.

[29] I. F. Akyldiz, "Wireless sensor network: A survey," *Comp. Networks J*, vol. 38, no. 4, pp. 393–422, 2002.

[30] D. Kandris, C. T. Nakas, D. Vomvas and G. E. Koulouras, "Applications of wireless sensor networks: An up-to-date survey," *Applied System Innovation*, 2020.

[31] M. Majid, S. Habib, A. R. Javed *et al.*, "Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review," *Sensors*, vol. 22, no. 6, p. 2087, 2022.

[32] A. Ali, Y. Ming, S. Chakraborty and S. Iram, "A comprehensive survey on real-time applications of wsn," *Future internet*, vol. 9, no. 4, p. 77, 2017.

[33] S. J. Ramson and D. J. Moni, "Applications of wireless sensor networks—a survey," in *2017 international conference on innovations in electrical, electronics, instrumentation and media technology (ICEEIMT)*, IEEE, 2017, pp. 325–329.

[34] R. Kashyap, "Applications of wireless sensor networks in healthcare," in *IoT and WSN applications for modern agricultural advancements: Emerging research and opportunities*, IGI Global, 2020, pp. 8–40.

[35] J. P. Mishra, K. Singh and H. Chaudhary, "Research advancements in ocean environmental monitoring systems using wireless sensor networks: A review," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 3, pp. 513–527, 2023.

[36] S. Duobiene and G. Račiukaitis, "Design of wireless sensor networks for environmental monitoring using lora," *International Journal of Environmental and Ecological Engineering*, vol. 17, no. 5, pp. 74–78, 2023.

[37] S. Sridharan, S. Venkatraman and S. Raja, "A novel lie hypergraph based lifetime enhancement routing protocol for environmental monitoring in wireless sensor networks," *IEEE Transactions on Computational Social Systems*, 2023.

[38] S. K. Swami Durai, B. Duraisamy and J. Thirukrishna, "Certain investigation on healthcare monitoring for enhancing data transmission in wsn," *International journal of wireless information networks*, vol. 30, no. 1, pp. 103–110, 2023.

[39] T. Jabeen, I. Jabeen, H. Ashraf, N. Jhanjhi, A. Yassine and M. S. Hossain, "An intelligent healthcare system using iot in wireless sensor network," *Sensors*, vol. 23, no. 11, p. 5055, 2023.

[40] D. Sharma and J. Singh, "A comparative analysis of healthcare monitoring systems using wsn," *ECS Transactions*, vol. 107, no. 1, p. 8695, 2022.

[41] M. M. Rahaman and M. Azharuddin, "Wireless sensor networks in agriculture through machine learning: A survey," *Computers and Electronics in Agriculture*, vol. 197, p. 106 928, 2022.

[42] P. K. Singh and A. Sharma, "An intelligent wsn-uav-based iot framework for precision agriculture application," *Computers and Electrical Engineering*, vol. 100, p. 107 912, 2022.

[43] M. Gheisari, M. S. Yaraziz, J. A. Alzubi *et al.*, "An efficient cluster head selection for wireless sensor network-based smart agriculture systems," *Computers and Electronics in Agriculture*, vol. 198, p. 107 105, 2022.

[44] J. Åkerberg, M. Gidlund and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *2011 9th IEEE International Conference on Industrial Informatics*, IEEE, 2011, pp. 410–415.

[45] D. Christin, P. S. Mogre and M. Hollick, "Survey on wireless sensor network technologies for industrial automation: The security and quality of service perspectives," *Future Internet*, vol. 2, no. 2, pp. 96–125, 2010.

[46] M. Paavola and K. Leiviska, "Wireless sensor networks in industrial automation," in *Factory Automation*, IntechOpen, 2010.

[47] T. M. Ghazal, M. K. Hasan, H. M. Alzoubi, M. Alshurideh, M. Ahmad and S. S. Akbar, "Internet of things connected wireless sensor networks for smart cities," in *The Effect of Information Technology on Business and Marketing Intelligence Systems*, Springer, 2023, pp. 1953–1968.

[48] S. W. Nourildean, M. D. Hassib and Y. Abd Mohammed, "Ad-hoc routing protocols in wsn-wifi based iot in smart home," in *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, IEEE, 2023, pp. 82–87.

[49] S. Mukhopadhyay and N. K. Suryadevara, *Smart cities and homes: Current status and future possibilities*, 2023.

[50] W. Osamy, A. M. Khedr, D. Vijayan and A. Salim, "Tactirso: Trust aware clustering technique based on improved rat swarm optimizer for wsn-enabled intelligent transportation system," *The Journal of Supercomputing*, vol. 79, no. 6, pp. 5962–6016, 2023.

[51] S. Kumar, S. Sharma and R. Kumar, "Wireless sensor network based real-time pedestrian detection and classification for intelligent transportation system," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 8, no. 2, p. 194, 2023.

[52] B. Epela, A. Manirabona and F. Nahayo, "Iitlma, an intelligent traffic light management algorithm based on wireless sensor networks," *Wireless Personal Communications*, pp. 1–11, 2023.

[53] C. V. Mahamuni, "A military surveillance system based on wireless sensor networks with extended coverage life," in *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*, IEEE, 2016, pp. 375–381.

[54] C. V. Mahamuni and Z. M. Jalauddin, "Intrusion monitoring in military surveillance applications using wireless sensor networks (wsns) with deep learning for multiple object detection and tracking," in *2021 International Conference on Control, Automation, Power and Signal Processing (CAPS)*, IEEE, 2021, pp. 1–6.

[55] M. P. Đurišić, Z. Tafa, G. Dimić and V. Milutinović, "A survey of military applications of wireless sensor networks," in *2012 Mediterranean conference on embedded computing (MECO)*, IEEE, 2012, pp. 196–199.

[56] M. Abdulkarem, K. Samsudin, F. Z. Rokhani and M. F. A Rasid, "Wireless sensor network for structural health monitoring: A contemporary review of

technologies, challenges, and future direction," *Structural Health Monitoring*, vol. 19, no. 3, pp. 693–735, 2020.

[57] A Sofi, J. J. Regita, B. Rane and H. H. Lau, "Structural health monitoring using wireless smart sensor network–an overview," *Mechanical Systems and Signal Processing*, vol. 163, p. 108 113, 2022.

[58] A. B. Noel, A. Abdaoui, T. Elfouly, M. H. Ahmed, A. Badawy and M. S. Shehata, "Structural health monitoring using wireless sensor networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1403–1423, 2017.

[59] H. Ma and Y. Liu, "Some problems of directional sensor networks," *Int. J. Sens. Networks*, vol. 2, pp. 44–52, 2007.

[60] S. Megerian, F. Koushanfar, G. Qu, G. Veltri and M. Potkonjak, "Exposure in wireless sensor networks: Theory and practical solutions," *Wireless Networks*, vol. 8, pp. 443–454, 2002.

[61] G. Veltri, Q. Huang, G. Qu and M. Potkonjak, "Minimal and maximal exposure path algorithms for wireless embedded sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 40–50.

[62] Y. Zou and K. Chakrabarty, "A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks," *IEEE Transactions on Computers*, vol. 54, pp. 978–991, 2005.

[63] 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, 2016, ISSN: 0045-7906.

[64] S. K. Gupta, P. Kuila and P. K. Jana, "Genetic algorithm for k-connected relay node placement in wireless sensor networks," in *Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015, Volume 1*, Springer, 2016, pp. 721–729.

[65] M. S., S. Udgata and S. Sabat, "M-connected coverage problem in wireless sensor networks," *ISRN Sensor Networks*, vol. 2012, Mar. 2012.

[66] A. K. Srivastava and S. K. Gupta, "Eerp: Energy-efficient relay node placement for k-connected wireless sensor networks using genetic algorithm," in *Ambient Communications and Computer Systems*, Y.-C. Hu, S. Tiwari, K. K. Mishra and M. C. Trivedi, Eds., Singapore: Springer Singapore, 2019, pp. 3–10, ISBN: 978-981-13-5934-7.

[67] Y.-h. Kim, C.-M. Kim, D.-S. Yang, Y.-j. Oh and Y.-H. Han, "Regular sensor deployment patterns for p-coverage and q-connectivity in wireless sensor networks," in *The International Conference on Information Network 2012*, 2012, pp. 290–295.

[68] D. Brélaz, "New methods to color the vertices of a graph," *Communications of the ACM*, vol. 22, no. 4, pp. 251–256, 1979.

[69] R. Wilkov and W. Kim, "A practical approach to the chromatic partition problem," *Journal of the Franklin Institute*, vol. 289, no. 5, pp. 333–349, 1970.

[70] C.-J. Tseng, "Automated synthesis of data paths in digital systems (design space)," Ph.D. dissertation, Carnegie Mellon University, 1984.

[71] F. T. Leighton, "A graph coloring algorithm for large scheduling problems," *Journal of research of the national bureau of standards*, vol. 84, no. 6, p. 489, 1979.

[72] N. J. Pullman, "Clique covering of graphs iv. algorithms," *SIAM Journal on Computing*, vol. 13, no. 1, pp. 57–75, 1984.

[73] N. Pullman, H Shank and W. Wallis, "Clique coverings of graphs v: Maximal-clique partitions," *Bulletin of the Australian Mathematical Society*, vol. 25, no. 3, pp. 337–356, 1982.

[74] R. Rees, "Minimal clique partitions and pairwise balanced designs," *Discrete mathematics*, vol. 61, no. 2-3, pp. 269–280, 1986.

[75] J. Bhasker and T. Samad, "The clique-partitioning problem," *Computers & Mathematics with Applications*, vol. 22, no. 6, pp. 1–11, 1991, ISSN: 0898-1221.

[76] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to algorithms*. MIT press, 2022.

[77] S. Mini, S. K. Udgata and S. L. Sabat, "Sensor deployment in 3-d terrain using artificial bee colony algorithm," in *International Conference on Swarm, Evolutionary, and Memetic Computing*, 2010.

[78] W. Chen, P. Yang, W. Zhao and L. Wei, "Improved ant lion optimizer for coverage optimization in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[79] R. Song, Z. Xu and Y. Liu, "Wireless sensor network coverage optimization based on fruit fly algorithm.," *International Journal of Online Engineering*, vol. 14, no. 6, 2018.

[80] A. N. Njoya, C. Thron, J. Barry *et al.*, "Efficient scalable sensor node placement algorithm for fixed target coverage applications of wireless sensor networks," *IET Wireless Sensor Systems*, vol. 7, no. 2, pp. 44–54, 2017.

[81] W. Wang, H. Huang, F. He, F. Xiao, X. Jiang and C. Sha, "An enhanced virtual force algorithm for diverse k-coverage deployment of 3d underwater wireless sensor networks," *Sensors*, vol. 19, no. 16, p. 3496, 2019.

[82] R. Özdağ and M. Canayaz, "Optimization of sensor deployment for k-coverage in wireless sensor networks," 2018.

[83] S. Afizudeen and R. Pavithra, *Grundy number based optimal sensor placement in 3d wireless sensor network*, 2023.

[84] A. Singh, A. Rossi and M. Sevaux, "Matheuristic approaches for q-coverage problem versions in wireless sensor networks," *Engineering Optimization*, vol. 45, no. 5, pp. 609–626, 2013.

[85] E. Doukhnitch, M. Salamah and E. Ozen, "An efficient approach for trilateration in 3d positioning," *Computer Communications*, vol. 31, no. 17, pp. 4124–4129, 2008, ISSN: 0140-3664.