

Review

A Survey of Three-Dimensional Wireless Sensor Networks Deployment Techniques

Tingting Cao ¹, Fan Yang ^{1,*} , Chensiyu Fan ¹, Ru Han ², Xing Yang ³  and Lei Shu ^{2,*} 

¹ School of Electrical Engineering and Automation, Jiangsu Normal University, Xuzhou 221116, China; caotingting@jsnu.edu.cn (T.C.); fanchensiyu@jsnu.edu.cn (C.F.)

² School of Smart Agriculture (Artificial Intelligence), Nanjing Agricultural University, Nanjing 210031, China; ruhan@stu.njau.edu.cn

³ College of Intelligent Manufacturing, Anhui Science and Technology University, Chuzhou 233100, China; xingyang@ahstu.edu.cn

* Correspondence: yangfan@jsnu.edu.cn (F.Y.); lei.shu@ieee.org (L.S.)

Abstract

Three-dimensional (3D) wireless sensor networks (WSNs) are gaining increasing significance in applications across complex environments, including underwater monitoring, mountainous terrains, and smart cities. Compared to two-dimensional (2D) WSNs, 3D WSNs introduce unique challenges in coverage, connectivity, map construction, and blind area detection. This paper provides a comprehensive survey of node deployment strategies in 3D WSNs. We summarize several key design aspects: sensing models, occlusion detection, coverage and connectivity, sensor mobility, signal and protocol effects, and simulation map construction. Deployment algorithms are categorized into six main types: classical algorithms, computational geometry algorithms, virtual force algorithms, evolutionary algorithms, swarm intelligence algorithms, and approximation algorithms. For each category, we review representative works, analyze their design principles, and evaluate their advantages and limitations. Comparative summaries are included to facilitate algorithm selection based on specific deployment requirements. Recent advancements in these strategies have led to significant improvements in network performance, with some algorithms achieving up to 12.5% lower cost and 30% higher coverage compared to earlier methods, and even reaching 100% coverage in certain cases. Thus, this survey aims to present the current research status and highlight practical improvements, offering a reference for understanding existing approaches and selecting appropriate algorithms for diverse deployment scenarios.

Keywords: three dimensional; wireless sensor networks; deployment algorithm; coverage; literature review



Received: 9 August 2025

Revised: 20 September 2025

Accepted: 22 September 2025

Published: 24 September 2025

Citation: Cao, T.; Yang, F.; Fan, C.; Han, R.; Yang, X.; Shu, L. A Survey of Three-Dimensional Wireless Sensor Networks Deployment Techniques. *J. Sens. Actuator Netw.* **2025**, *14*, 94. <https://doi.org/10.3390/jsan14050094>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

WSNs consist of a large number of micro sensor nodes, and each sensor node typically comprises a sensing component, power supply, processor, and communication module. The sensing component detects environmental parameters such as temperature, humidity, sound, and pressure; the processor handles data processing and local decision-making; and the communication module ensures wireless data transmission. The power supply, often a battery or energy-harvesting module, governs the lifetime of nodes. These components enable nodes to collaboratively perceive and monitor environmental target information and transmit data wirelessly to sink nodes. Generally, the effectiveness of sensor nodes is

largely determined by two key parameters: sensing range and communication range. The sensing range defines a node's capability to detect target events, while the communication range governs its ability to transmit data to other nodes [1].

At the network level, WSNs are characterized by distributed sensing, multi-hop communication, self-organization, and scalability. The nodes collectively form a dynamic network topology, enabling robust data collection, fault tolerance, and efficient energy management. These technical features allow WSNs to perform large-scale monitoring tasks in complex environments. Additionally, quality of service (QoS) metrics such as coverage, latency, reliability, and network lifetime are essential for evaluating WSN performance and guiding deployment strategies. Ensuring adequate QoS requires the careful consideration of both node capabilities and network design [2].

With advancements in sensor hardware and communication technologies, WSNs have found widespread applications across diverse fields, including environmental monitoring [3], smart agriculture [4], smart cities [5], disaster warning [6], underwater detection [7], industrial safety [8], and healthcare [9]. Additionally, the integration of WSNs with mobile cloud computing enhances the data processing and scalability of these applications [10]. In these applications, node deployment emerges as a critical factor influencing network performance, directly impacting coverage, connectivity, energy efficiency, and service lifetime.

WSN nodes can be deployed in 2D spaces, 3D spaces, or on 3D surfaces. Two-dimensional deployment involves distributing nodes on flat surfaces, a common scenario in research on island [11] and sand terrains [12]. For 3D deployment, the current research primarily focuses on two scenarios: one involves deploying nodes within 3D volumes (e.g., underwater sensing [13–16] and indoor monitoring [17–19]), where spherical sensing models are typically used to fill the space; the other involves deploying nodes on complex 3D surfaces (e.g., mountainous terrains [20,21]), where occlusion and visibility must be considered. Figure 1 illustrates the conceptual differences between 2D deployment, 3D space deployment, and 3D surface deployment.

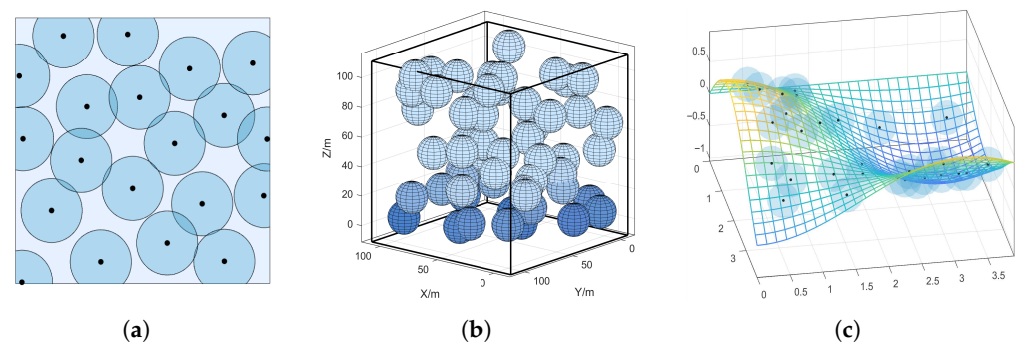


Figure 1. The 2D vs. 3D space vs. 3D surface deployment: (a) 2D deployment, (b) 3D space deployment, (c) 3D surface deployment.

Traditional 2D WSN deployment methods [2] have been widely adopted owing to their simplified modeling and ease of implementation. Nevertheless, models designed for 2D WSNs are no longer applicable to 3D WSNs, primarily due to the unique constraints introduced by 3D spaces. For instance, line of sight (LoS) emerges from visual height differences, a phenomenon that does not occur in 2D scenarios. In comparison with 2D scenarios, 3D deployments present more pronounced challenges, including larger spatial volumes, more intricate communication protocols, higher costs, and greater difficulties in localization [22,23]. Furthermore, the QoS requirements in 3D WSNs differ significantly from 2D WSNs. While 2D WSNs mainly focus on planar coverage and relatively simple routing, 3D WSNs must address volumetric coverage and surface coverage, complex multi-hop routing, increased interference, and potentially higher latency, all of which directly impact

reliability and network lifetime [24,25]. To address the unique challenges of 3D deployment, various WSN models are employed in 3D WSNs. Commonly used sensing models in 3D deployment include binary models with fixed sensing ranges, probabilistic models that account for environmental uncertainty, and directional models designed for sensors with limited fields of view. Sensor nodes may be static, mobile, or exhibit hybrid mobility to adapt to spatial dynamics. In terms of system architecture, centralized approaches rely on a central controller to gather global information and make deployment decisions, whereas distributed architectures enable individual nodes to make decisions based on local data, thereby enhancing adaptability and scalability. Signal propagation and communication protocols jointly shape link reliability, coordination mechanisms, and energy efficiency in 3D WSNs. Connectivity requirements range from none to 1-connectivity (which ensures basic network functionality) and m -connectivity (which guarantees that each node is connected to at least m neighbors to improve fault tolerance). Coverage goals are equally diverse: 1-coverage ensures that each target point is covered by at least one sensor; K -coverage requires all target points to be covered by at least K sensors for redundancy; and Q -coverage generalizes this concept by assigning different coverage degrees to different points based on their sensing importance. To better understand the fundamental elements of 3D WSNs, we summarize the key WSN models in Figure 2.

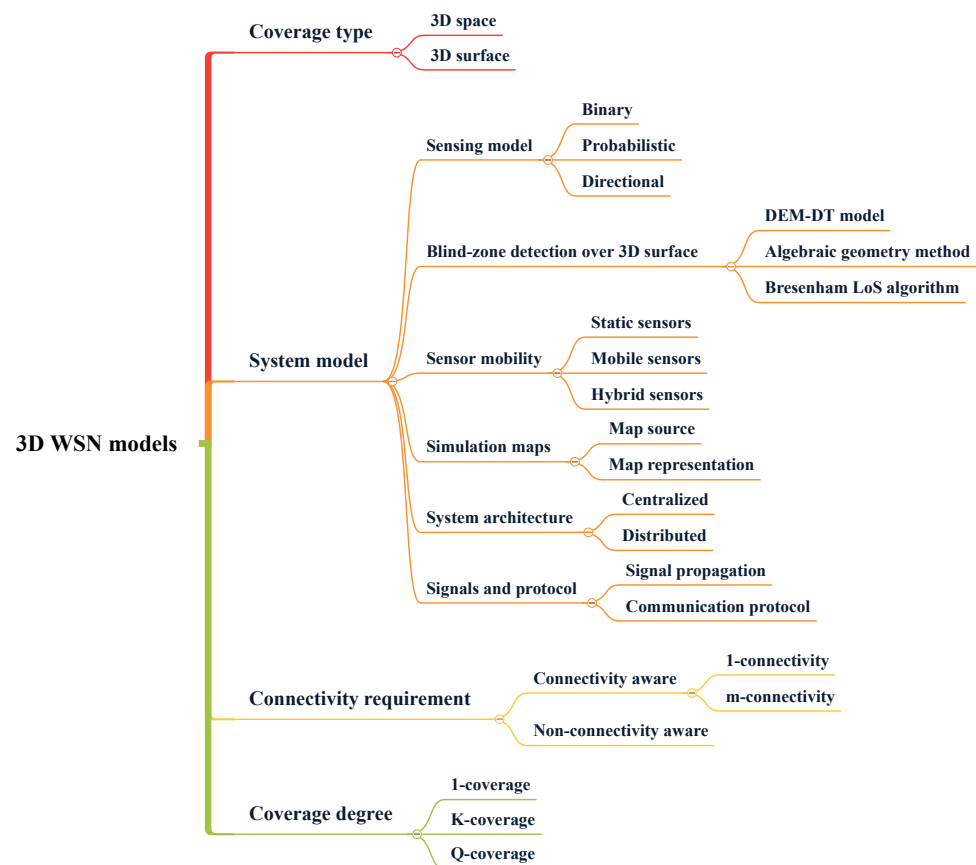


Figure 2. Common deployment models in 3D WSNs.

In recent years, numerous researchers have made significant contributions to the study of node deployment in 3D WSNs. For example, the authors of [26] systematically analyzed various coverage models and computational methods, considering factors such as sensor detection range, sensing angles, terrain visibility, weather, and obstacles. They identified key limitations in existing studies, including the misuse of terrain models (e.g., applying grid models to complex terrains), unrealistic assumptions (e.g., neglecting terrain

occlusion or assuming omnidirectional sensing), and a lack of experimental validation. However, their work did not discuss deployment algorithms that are commonly used in 3D WSNs. Focusing on indoor 3D deployment research, the authors of [27] introduced relevant concepts, types, objectives, models, and application areas. They further explored critical issues in IoT data collection networks, such as coverage, localization, connectivity, network lifetime, and data traffic, while presenting methods to address indoor 3D deployment challenges. Nevertheless, this study insufficiently explored deployment scenarios on 3D surfaces. In [28], the authors reviewed and compared five typical optimization algorithms, i.e., greedy algorithm, genetic algorithm (GA), particle swarm optimization (PSO), differential evolution (DE), and binary integer programming (BIP), for enhancing the visual coverage of 3D objects. Their research primarily targeted 3D surface coverage in visual sensor networks. However, it overlooked 3D spatial deployment and key WSN metrics such as connectivity and mobility. The authors of [29] summarized the wide-ranging applications of 3D WSNs, analyzed differences in node types, sensing models, deployment methods, and mobility, and briefly reviewed deployment approaches along with their associated models and optimization objectives. The paper provided a detailed introduction to polyhedron-filling deployment methods. Yet, it lacked a comprehensive classification and discussion of various deployment algorithms, and insufficiently introduced relevant system models in WSNs.

Despite the existence of multiple surveys on node deployment in 3D WSNs, most existing reviews have notable shortcomings. Many focus on a single coverage type (either 3D surfaces or 3D spaces) without comprehensively analyzing both. Additionally, they fail to thoroughly summarize diverse WSN characteristics and deployment algorithms, lack detailed explanations of the sources and representations of simulation maps, omit discussions on the Q-coverage problem in coverage studies, and lack systematic analysis in comparative evaluations of algorithms. To address these limitations, this paper presents a comprehensive and structured survey of 3D WSN deployment strategies. Our goal is to bridge existing research gaps by reviewing a broad range of algorithms and analyzing their performance from the perspectives of coverage, connectivity, mobility, and blind-zone detection. We also emphasize the importance of simulation map design and assess algorithm applicability in various 3D surface and space settings. This work aims to provide a practical reference for researchers and practitioners seeking effective deployment solutions in complex 3D environments. Table 1 highlights the key differences between the aforementioned reviews and this study.

Table 1. Comparison between the related reviews and our review.

Reference	Year	3D Space	3D Surface	Algorithm	Binary	Probabilistic	Directional	Blind-Zone	Sensor Mobility	Map Source	Map Representation	Connectivity	K-Coverage	Q-Coverage	Distributed	Centralized
[26]	2019		✓		✓	✓	✓	✓			✓					
[27]	2019	✓	✓	✓	✓	✓			✓			✓	✓			
[28]	2019		✓	✓			✓	✓								
[29]	2019	✓	✓	✓	✓	✓			✓			✓	✓		✓	✓
This Paper	2025	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Motivated by the aforementioned discussions and to provide an up-to-date overview, this paper mainly reviews articles published in the past ten years, with emphasis on recent studies. Most focus on node deployment in 3D environments, and we aim to cover a wide range of approaches to ensure the survey is comprehensive. The paper then

conducts a comprehensive survey on the main challenges and optimization methods of node deployment in 3D WSNs, with a focus on key design aspects such as coverage and connectivity, energy management, node mobility, and map source and representation. The primary contributions of this paper are summarized as follows:

1. We summarize five key design elements for 3D WSN deployment: sensing models, occlusion handling for 3D surfaces, coverage and connectivity, sensor mobility, signal and protocol effects, and simulation maps. These elements form the fundamental framework for understanding and addressing 3D deployment issues.
2. We systematically classify existing methods into six categories according to algorithm design concepts: classical algorithms, computational geometry methods, virtual force methods, evolutionary algorithms, swarm intelligence algorithms, and approximation algorithms. For each category, their strengths and limitations in different deployment scenarios are summarized, providing a clear reference for algorithm selection.
3. We offer comparative tables of representative deployment strategies. These tables help researchers quickly identify and select appropriate optimization methods based on specific application requirements, enhancing the practicality of the survey.
4. We present a case study on the deployment of solar insecticidal lamps (SILs) in complex 3D agricultural environments. This case highlights the impact of terrain variability, occlusion, solar exposure, and accessibility constraints on practical deployment strategies and the design of optimization models, bridging the gap between theoretical research and real-world applications.

To facilitate readers' understanding, the acronyms used in this paper are listed in Table 2 and notations are listed in Table 3.

Table 2. Acronyms in this paper.

Acronym	Description	Acronym	Description
WSNs	Wireless Sensor Networks	FOA	Fruit Fly Optimization Algorithm
3D	Three Dimensional	BHA	Black Hole Algorithm
DEM	Digital Elevation Model	MPA	Marine Predator Algorithm
LoS	Line-of-Sight	CSA	Cuckoo Search Algorithm
TIN	Triangulated Irregular Networks	BOA	Butterfly Optimization Algorithm
CG	Computational Geometry	GWO	Grey Wolf Optimizer
VD	Voronoi Diagram	SSA	Sparrow Search Algorithm
DT	Delaunay Triangulation	LSOA	Lion Swarm Optimization Algorithm
VF	Virtual Force	CLO	Competitive Learning Optimizer
EA	Evolutionary Algorithm	PTSA	Parallel Tunicate Swarm Algorithm
GA	Genetic Algorithm	AFSA	Artificial Fish Swarm Algorithm
DE	Differential Evolution	BOA	Bat Optimization Algorithm
SI	Swarm Intelligence	SFLA	Shuffled Frog Leaping Algorithm
PSO	Particle Swarm Optimization	WOA	Whale Optimization Algorithm

Table 3. Notations in this paper.

Notation	Description	Notation	Description
$\mathcal{S} = \{s_i\}_{i=1}^m$	Collection of sensor nodes	$\mathcal{P} = \{p_j\}_{j=1}^n$	Collection of target points
$d(s, p)$	Distance between s and p	P_{ij}	Probability that p_j can be covered by s_i
R_s	Sensing radius of sensor	R_c	Communication radius of sensor
P_s	Decision variable for blind spot	z_p	Elevation value of point p
$ \mathcal{S} $	Cardinality of collection $\mathcal{S}$	QoC	Quality of coverage
$QCon$	Quality of connectivity	d_{th}	Distance threshold in VF algorithm

The rest of this paper is organized as follows: Section 2 introduces the basic design elements of 3D deployment; Section 3 summarizes six typical algorithm categories and their optimization models; Section 4 reviews the current applications of various algorithms in 3D deployment; Section 5 compares the deployment effectiveness and applicability of

different algorithms; Section 6 analyzes a case study on the deployment of solar insecticidal lamps in a 3D environment; and Section 7 concludes the paper and outlines future research directions.

2. Fundamental Models in 3D WSNs

2.1. Sensing Models

In WSNs, the Euclidean distance is commonly used to represent the geometric distance between a sensor node and a point in the target region. Given a sensor s located at (x_s, y_s, z_s) and a target point p at (x_p, y_p, z_p) , the distance $d(s, p)$ between s and p is defined as follows:

$$d(s, p) = \sqrt{(x_s - x_p)^2 + (y_s - y_p)^2 + (z_s - z_p)^2} \quad (1)$$

This measure serves as the basis for evaluating proximity in many coverage and localization models. While it exhibits certain inaccuracies in characterizing communication paths, the Euclidean distance model can still approximately describe how sensing probability varies along the propagation direction. However, when deploying sensors on 3D surfaces such as mountains and volcanoes, the actual distance between two points should be the shortest path along the surface, known as the geodesic distance, rather than the Euclidean distance. Geodesic distance provides a way to assess visibility without explicitly determining LoS. Unlike Euclidean distance, geodesic distance is defined as the shortest path along the surface or obstacle terrain, naturally incorporating terrain undulations and occluding objects [30].

Figure 3 illustrates a simple example that visually compares the Euclidean distance (blue dashed line, from A to B in a straight line) with the geodesic distance (red curve, the shortest path from A to B along the surface).

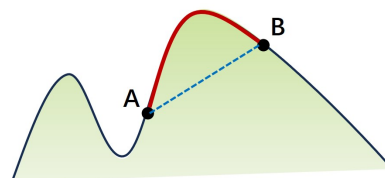


Figure 3. The Euclidean distance (blue dashed line) vs. the geodesic distance (red solid line).

Based on the distance calculation method, sensing models in WSNs can be categorized into the binary sensing model, probabilistic sensing model, and directional sensing model. In 3D space, these sensing models are typically used according to different requirements and application scenarios. In contrast, for 3D surfaces, LoS constraints are often combined with the sensing models (as discussed in [31]), or geodesic distance is employed instead of the traditional Euclidean distance (as explained in [32]). For details on LoS determination, see Section 2.2; for a comparison of sensing models and constraints in 3D space and on 3D surface, refer to Table 4.

Table 4. Sensing models and constraints in 3D space and surface.

Coverage Type	Case	Sensing Model	Distance Type	LoS Constraint
3D Space	case 1	binary, probabilistic, directional	Euclidean distance	No
3D Surface	case 1	binary, probabilistic, directional	Euclidean distance	Yes
	case 2	binary, probabilistic, directional	geodesic distance	No

2.1.1. Binary Sensing Model

The binary sensing model is a common model in WSNs. It assumes that a sensor can perfectly detect any point located within a fixed sensing radius R_s but cannot detect points

outside this range. For a target point p in the region of interest (RoI), the probability that p is covered by sensor s , denoted as P_{ij} , can be expressed as

$$P_{ij} = \begin{cases} 1, & \text{if } d(s, p) \leq R_s \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

This binary sensing model provides a deterministic and computationally simple framework for coverage analysis. However, in practical applications, it fails to accurately reflect the sensing characteristics of sensor nodes due to factors such as external noise, obstacles, and the attenuation of wireless signal strength with increasing distance. This model is illustrated in Figure 4a.

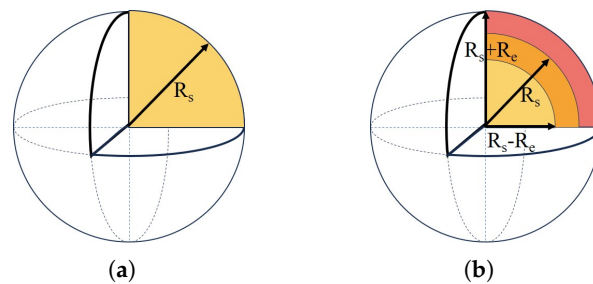


Figure 4. Binary sensing model and probabilistic sensing model. (a) Binary sensing model. (b) Probabilistic sensing model.

To better capture the complexity of real-world environments, some studies have proposed enhancements to the traditional binary sensing model. For instance, the authors in [33] adopt an ellipsoidal sensing region instead of a circular one. This modification more accurately reflects signal propagation characteristics in practical scenarios, e.g., anisotropic attenuation or directional sensor properties, thereby improving the realism and effectiveness of sensor deployment strategies.

2.1.2. Probabilistic Sensing Model

The probabilistic sensing model is a more realistic approach in wireless sensor networks, addressing the limitations of the idealized binary model by incorporating uncertainty in detection. Unlike the deterministic binary model, it accounts for signal attenuation, environmental noise, and obstacles by defining sensing probability as a continuous function of distance. The computational method of this model can be expressed as

$$P_{ij} = \begin{cases} 1, & \text{if } d(s, p) \leq R_s - R_e \\ e^{-\alpha(d(s, p) - R_s)^\beta}, & \text{if } R_s - R_e < d(s, p) < R_s + R_e \\ 0, & \text{if } d(s, p) \geq R_s + R_e \end{cases} \quad (3)$$

where R_e represents the uncertainty range of the sensing boundary, and α and β control the decay rate and shape of the sensing probability curve near the boundary, respectively. Within the range $R_s - R_e$, the sensor can reliably detect targets: in the interval between $R_s - R_e$ and $R_s + R_e$, the sensing probability decreases exponentially with increasing distance; beyond $R_s + R_e$, the sensing probability drops to zero. This model is shown in Figure 4b.

However, some studies have simplified the traditional probabilistic sensing model [34,35] as defined in Equation (4). This simplified model replaces the uncertain sensing boundary interval with a fixed radius R_s , where λ is a parameter related to the physical characteristics of the sensor node. It removes the gradual probability decay and exponential function,

using a binary sensing judgment instead. This improves computational efficiency but results in lower realism:

$$P_{ij} = \begin{cases} e^{-\lambda \cdot d(s,p)} & \text{if } d(s,p) \leq R_s \\ 0 & \text{if } d(s,p) > R_s \end{cases} \quad (4)$$

2.1.3. Directional Sensing Model

Directional sensing models describe the limited field of view and sensing range of sensors using 3D geometric shapes such as pyramids, cones, or spherical sectors, distinguishing them from traditional omnidirectional models. These models precisely characterize the sensing direction and area of nodes through parameters like horizontal angle, vertical angle, and orientation. They can also incorporate distance and angle attenuation to construct probabilistic sensing functions, and some models allow adjustment of sensing direction to adapt to environmental and task requirements. Such features reflect the actual working conditions of sensors and optimize node deployment and coverage in 3D space.

Different studies have adopted various directional sensing models. In [36], the sensing range of a video sensor is modeled as a pyramid (as illustrated in Figure 5a), similar to a camera's view frustum, with the trapezoidal monitoring area determined by parameters such as horizontal field of view, vertical field of view, and tilt angle. Considering that objects may not be clearly imaged if too close to the camera, the authors in [17] define the sensing range as a frustum-shaped pyramid. In [37], a pyramid perpendicular to the horizontal plane is used as the sensor's sensing model. The authors in [38] propose a 3D cone directional sensing model (as illustrated in Figure 5b) by integrating target height information and sensor sensing radius; its range is determined using conical geometry and directional parameters, with coverage varying according to pitch and deflection angles. To adjust the sensing range more efficiently, the authors in [39] apply a spherical sector model (as shown in Figure 5c) to represent the directional sensing range of a sensor node.

In addition, other studies such as [40–43] also employ various 3D directional sensing models to more accurately describe the spatial sensing characteristics of directional sensors operating in complex and heterogeneous environments. These models often consider factors like limited sensing angles, coverage range variations with distance, and the impact of terrain or obstacles, enabling more precise modeling of sensor behavior in 3D space.

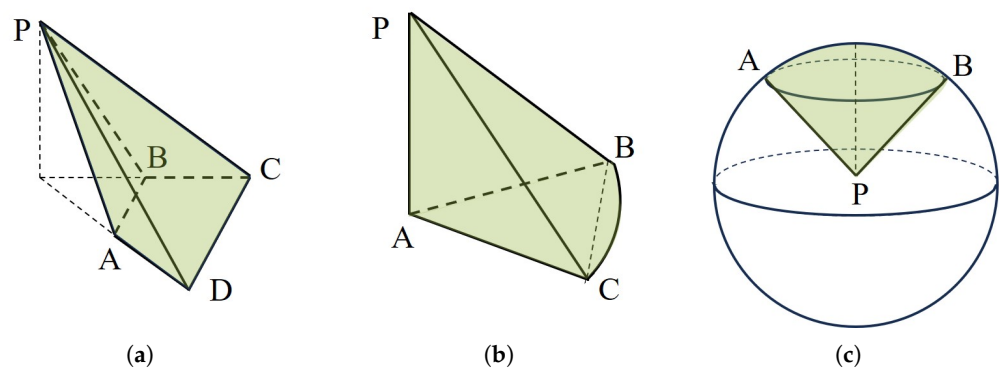


Figure 5. Directional sensing models. (a) Pyramid sensing model. (b) Cone sensing model. (c) Spherical sector sensing model.

2.2. Blind Zone Detection over 3D Surfaces

Occlusions on 3D surfaces create blind spots for sensor monitoring, which can significantly impair the accuracy and effectiveness of sensor nodes. Common methods for blind spot detection include the digital elevation model combined with the Delaunay triangulation (DEM-DT) model, algebraic geometry approaches, and the Bresenham LoS algorithm.

2.2.1. DEM-DT Model

The blind spot detection method proposed in [31] combines the digital elevation model (DEM) with Delaunay triangulation (DT). Its core idea is to transform complex terrain surfaces into multiple triangular regions using DT, deploy sensor nodes at the circumcenters of these triangles, and set the sensing radius to cover all vertices. An example of the DEM-DT approach is shown in Figure 6.

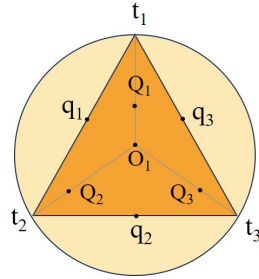


Figure 6. An example of the DEM-DT model.

In this example, $\mathcal{T} = \{t_1, t_2, t_3\}$ represent the three vertices of the DT with elevation information, O_1 denotes the circumcenter of the DT, and $\mathcal{Q}^* = \{q_1, q_2, q_3\}$ and $\mathcal{Q} = \{Q_1, Q_2, Q_3\}$ are interpolation points calculated based on the elevation values at t_1, t_2, t_3 . Let z_p denote the elevation values of point $p \in \mathcal{T} \cup \mathcal{Q} \cup \mathcal{Q}^*$. Hence, the calculation formulas for these interpolation points and elevation values are given by Equations (5)–(11):

$$z_{q_1} = \frac{z_{t_1} + z_{t_2}}{2} \quad (5)$$

$$z_{q_2} = \frac{z_{t_1} + z_{t_3}}{2} \quad (6)$$

$$z_{q_3} = \frac{z_{t_2} + z_{t_3}}{2} \quad (7)$$

$$z_{O_1} = 0.2 \cdot \frac{z_{t_1} + z_{t_2} + z_{t_3}}{3} + 0.8 \cdot \frac{z_{q_1} + z_{q_2} + z_{q_3}}{3} \quad (8)$$

$$z_{Q_1} = 0.5 \cdot \frac{z_{t_1} + z_{O_1}}{2} + 0.5 \cdot \frac{z_{q_1} + z_{q_2}}{2} \quad (9)$$

$$z_{Q_2} = 0.5 \cdot \frac{z_{t_2} + z_{O_1}}{2} + 0.5 \cdot \frac{z_{q_1} + z_{q_3}}{2} \quad (10)$$

$$z_{Q_3} = 0.5 \cdot \frac{z_{t_3} + z_{O_1}}{2} + 0.5 \cdot \frac{z_{q_2} + z_{q_3}}{2} \quad (11)$$

The rule for judging whether a blind spot exists in the region where a sampling point is located is as follows: If the elevation value z_Q of a sampling point $Q \in \mathcal{Q}$ exceeds any vertex elevation z_t and also exceeds the circumcenter elevation z_{O_1} (as shown in Equation (12)), then a blind spot exists in this region; otherwise, the region is considered fully covered. Let P_s denote the decision variable for the blind spot, and thus P_s can be derived as follows:

$$P_s = \begin{cases} 1, & \forall Q \in \mathcal{Q}, \forall t \in \mathcal{T}, z_Q \leq z_t \\ 1, & \forall Q \in \mathcal{Q}, \forall t \in \mathcal{T}, z_t \leq z_Q \leq z_{O_1} \\ 0, & \exists Q \in \mathcal{Q}, \exists t \in \mathcal{T}, z_t < z_{O_1} \text{ and } z_{O_1} < z_Q \end{cases} \quad (12)$$

Thus, by combining the sensing models in (2) or (3) with Equation (12), the improved model can detect blind zones and be applied to 3D surfaces with occlusions. In this combined

model, P_{ij} is the traditional sensing probability, and P_s is the decision variable for the blind spot. The combined sensing model is expressed in Equation (13):

$$P = P_{ij} \cdot P_s \quad (13)$$

2.2.2. Algebraic Geometry Method

Algebraic geometry methods involve creating equations for the lines connecting sensor nodes and target points. These equations are then used to check for intersections with surfaces described by functions like $z = f(x, y)$. If an intersection is detected, it signals potential occlusion.

To better illustrate this algebraic geometry-based approach, a representative example is provided in Figure 7, which clearly shows how to determine the precise intersection point between a LoS and a surface [44,45]. It emphasizes the geometric calculations involved and their practical application in sensing and coverage analysis.

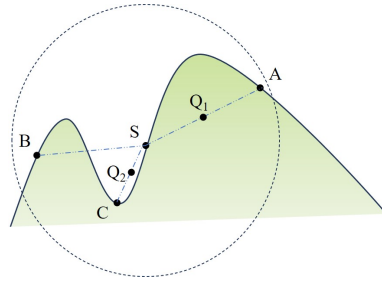


Figure 7. An example of the algebraic geometry method.

Let (x_s, y_s, z_s) , (x_a, y_a, z_a) , (x_b, y_b, z_b) , and (x_c, y_c, z_c) denote the coordinates the sensor node S and the three target points A, B, and C, respectively. Q_1 and Q_2 are the midpoints of line segments $\overline{AS}$ and $\overline{CS}$, respectively. The terrain surface is modeled as $z = f(x, y)$. Equation (14) represents the intersection between spatial line segment $\overline{BS}$ and the 3D surface $z = f(x, y)$. If Equation (14) has at least one solution, it indicates an obstacle exists between the two points, and target point B cannot be covered by sensor S:

$$\begin{cases} \frac{x - x_s}{x_b - x_s} = \frac{y - y_s}{y_b - y_s} = \frac{z - z_s}{z_b - z_s} \\ z = f(x, y) \end{cases} \quad (14)$$

Equation (15) represents the height relationship between midpoint $Q_1(x_q, y_q, z_q)$ of line segment $\overline{AS}$ and the 3D surface. If the elevation value z_{Q_1} of Q_1 , is less than the corresponding value of the surface function, there is an obstruction at this point, and target point A cannot be covered by sensor S:

$$P_{\text{cov}}(S, A) = \begin{cases} 0 & \text{if } z_{Q_1} \leq f(x_q, y_q) \\ 1 & \text{otherwise.} \end{cases} \quad (15)$$

A target point is covered by a sensor only when the following three conditions are met simultaneously: (1) the sensing probability $P_{ij} = 1$; (2) Equation (14) has no solution; and (3) the value of Equation (15) equals 1.

2.2.3. Bresenham LoS Algorithm

The Bresenham LoS algorithm is frequently employed for occlusion analysis in 3D surfaces, especially suitable for blind spot detection in DEM or discrete terrain environments. The algorithm functions by discretely “drawing” a straight line between the sensor

node and the target point. It then iterates through all grid cells along this path to check if any obstacle's height exceeds the current height of the straight line so as to determine the existence of a LoS.

Figure 8 presents a simple LoS scenario. For the sensor s and target points p_1 and p_2 , if the height of every corresponding grid between them does not exceed the spatial line segment $\overline{sp_1}$, it means there is no obstacle between the sensor and p_1 , i.e., a LoS exists between s and p_1 . Conversely, there is no LoS between s and target point p_2 [46].

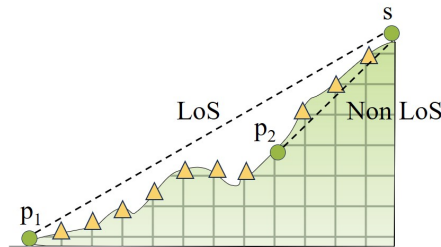


Figure 8. A Bresenham LoS scenario.

For a target point to be covered by a sensor, two conditions must be met: (1) the sensing probability $P_{ij} = 1$; and (2) a LoS must exist between the sensor and the target point. The Bresenham LoS algorithm has been applied in blind spot detection on 3D surfaces in studies like [47–49].

2.3. Coverage and Connectivity

In 3D WSNs, coverage and connectivity are two core design elements that determine network performance and reliability. A well-planned node deployment must balance both to ensure effective monitoring of the target area while enabling stable data transmission among nodes. However, the complex spatial distribution of nodes in 3D environments makes optimizing coverage and connectivity extremely challenging.

2.3.1. Coverage

Coverage in WSNs is categorized based on sensing redundancy into 1-coverage, K -coverage, and Q -coverage. These classifications guide the design and optimization of sensor placement to meet specific coverage objectives.

1-coverage requires every target point in the monitored area to be covered by at least one sensor node. With minimal redundancy, this model is ideal for applications with limited resources or low fault-tolerance needs. Its simplicity makes it widely used in initial deployment planning, basic layout optimization, and blind spot elimination. K -coverage enhances reliability by mandating each point to be covered by at least K sensors, improving redundancy and fault tolerance. For example, in cultural heritage site microclimate monitoring [50], robust K -coverage over complex 3D surfaces is critical. Studies like [14,32,51,52] have proposed optimization strategies to achieve effective 3D K -coverage. Q -coverage introduces flexibility by covering points with varying numbers of sensors based on their importance, supporting prioritized and hierarchical monitoring. Approaches in [53,54] dynamically adjust coverage levels to optimize resource allocation and boost reliability in 3D scenarios.

Coverage performance, a key metric for evaluating how well the monitored region is covered, reflects the spatial effectiveness and sensing reliability of the network. A more comprehensive metric is the quality of coverage (QoC) [55], defined as

$$QoC = \frac{|X(S)|}{n} \quad (16)$$

where $X(\mathcal{S}) = \{p \in \mathcal{P} \mid \exists s \in \mathcal{S}, d(s, p) \leq R_s\}$ denotes the set of target points that can be covered by the deployed sensors $\mathcal{S}$, and n is the total number of monitored target points.

2.3.2. Connectivity

Connectivity is vital for network integrity and efficient information transmission in WSNs, enabling communication between nodes and enhancing fault tolerance and robustness. It is classified by connectivity degree into 1-connectivity and m -connectivity.

1-connectivity ensures at least one path exists between any two sensor nodes, maintaining network connectivity as long as such paths persist. m -connectivity requires at least m node-disjoint paths between any pair of nodes. In graph theory terms, a network is m -connected if removing $(m-1)$ nodes does not disconnect the graph [56]. This property inherently provides fault tolerance, ensuring that the network remains connected even if up to $(m-1)$ nodes fail or are compromised. Fault tolerance is critical in 3D WSNs, as nodes may fail due to energy depletion, hardware faults, or environmental hazards. Maintaining multiple independent paths through m -connectivity ensures the network continues to work, supporting reliable data collection and communication in harsh or dynamic environments. To improve reliability, studies like [40] require each sensor and relay node to connect to at least two relays, preventing communication failures from node energy depletion.

The quantitative assessment of connectivity uses metrics such as quality of connectivity (QCon) [57]:

$$QCon = \frac{1}{m} \sum_{i=1}^m S_i \quad (17)$$

where m is the number of deployed sensors, and S_i is a decision variable used to determine whether there is a path between a node s_i and a base station (binary: 1 for successful communication, 0 otherwise). Other metrics include the proportion of isolated nodes (nodes unable to communicate with others or the base station) used in [58], and the average node degree from [59]:

$$QCon = \frac{1}{m} \sum_{i=1}^m Deg_i \quad (18)$$

where Deg_i is the number of neighbors of node S_i . Additionally, the authors in [33,60] define the connectivity ratio as the ratio of existing communication paths to the theoretical maximum.

2.4. Sensor Mobility

In complex 3D environments such as mountainous terrain and underwater regions, sensor nodes are often initially deployed randomly due to deployment constraints, resulting in coverage holes. Introducing mobile nodes for post-deployment adjustments can enhance the network's overall coverage and robustness. Compared to static WSNs, mobile WSNs offer higher adaptability and flexibility, making them widely applicable in monitoring, surveillance, search and rescue, and exploration of dangerous environments.

Based on node mobility, deployment strategies can be categorized into fully static, fully mobile, and hybrid deployments. Fully static deployment features a simple structure and low energy consumption but has limited adaptability to environmental changes. Fully mobile deployment allows for dynamic coverage adjustments but incurs high costs and energy consumption. For example, the authors in [61] employed mobile robots to achieve coverage in 3D space and reduce mobile energy consumption and extend lifetime through mobile optimization strategies and pre-designed mobile paths. Hybrid deployment strikes a balance between energy efficiency and adaptability because they confine mobility to a subset of nodes, thereby reducing the movement overhead while still retaining adaptability,

which prolongs operational lifetime compared to fully mobile systems. In [62], both static and mobile nodes are randomly deployed on a 3D surface. After detecting coverage holes, only redundant mobile nodes near the holes are moved, reducing unnecessary energy consumption. Additionally, mobile nodes are required to maintain at least one neighbor connection during relocation to preserve network connectivity. A similar approach is used in [51] to balance coverage and energy efficiency.

In mobility-based deployments, energy consumption is dominated by node movement, which requires continuous mechanical or propulsion effort and rapidly depletes batteries. As a result, maximizing network lifetime relies on minimizing unnecessary relocations and optimizing movement trajectories. Since sensor nodes operate with limited energy, especially in harsh environments, efficient energy management is critical; otherwise, node failures due to energy depletion may disconnect the network and disrupt data transmission. In [62], the maximum moving distance d_{max} for a node is constrained by its remaining energy E_{res} , expressed as

$$d_{max} = \sqrt[n]{\frac{E_{res}}{k}} \quad (19)$$

where k is a constant representing energy consumption per unit distance, and n is the exponent depending on the energy model. This formulation ensures energy-aware movement and prolongs the network's operational lifetime.

2.5. Signal and Protocol Effects in 3D WSNs

Signal propagation characterizes the properties of the radio channel, which are closely related to environmental and physical parameters. By predicting radio frequency signal attenuation and distortion, the expected received signal strength can be estimated. Sensors are generally assumed to be distributed on the same plane in 2D WSNs, where signal propagation is mainly affected by loss of free space and minor terrain reflections, resulting in relatively simplified models. In contrast, sensors deployed in complex environments, such as high-rise buildings, forests, or mountainous environments, exhibit significantly more complex propagation due to multipath effects and obstacle shadowing [63].

In 3D environments, the received signal strength indicator (RSSI) is commonly used to evaluate link accessibility, with its relationship to the transmit power expressed by the path-loss model [33]:

$$PowR = PowT \cdot \alpha \cdot u^{-\Phi} \quad (20)$$

where $PowT$ denotes the transmission power, $PowR$ denotes the reception power, u is the distance between nodes, Φ is the path loss exponent, and α is an environment-dependent constant. In practice, RSSI is often considered together with the frame error rate (FER) to determine reliable neighboring links. For example, RSSI must be above a threshold and FER below 0.4 to ensure link. The study of terrain measurement by the authors in [63] shows that low-altitude nodes in complex terrains experience propagation losses far beyond the predictions of 2D models, demonstrating the need for refined modeling for 3D scenarios.

Antenna directivity and interference also have a considerable impact on signal propagation. The authors in [64] demonstrate that optimizing antenna downtilt can significantly improve coverage probability and suppress inter-cell interference in heterogeneous 3D cellular networks. Similarly, the authors in [65] demonstrate through a three-dimensional ray-tracing framework that obstacles, building structures, and multipath propagation profoundly affect the performance of low-power networks such as LoRaWAN.

Signal propagation characteristics directly affect protocol design. Protocols determine how nodes manage channel access, data transmission, and energy consumption, all of which are shaped by the propagation environment. For instance, in 3D scenarios with stronger multipath and interference, protocols need to include more robust access control and error

recovery mechanisms to maintain reliability and energy efficiency. Among existing solutions, IEEE 802.15.4 is the most widely used standard for short-range, low-power communication, and has been extensively adopted in WSN deployments [66].

In summary, signal propagation models and communication protocols jointly influence the performance of WSNs, including coverage, connectivity, and energy consumption. Therefore, selecting appropriate propagation models and protocol mechanisms is critical to enhancing the performance and reliability of deployment algorithms in complex 3D scenarios.

2.6. Simulation Maps for Node Deployment in 3D WSNs

Simulation maps play a crucial role in node deployment, as they provide representations of terrain features essential for optimizing sensor placement, coverage, and connectivity in real-world applications. They are often supported by simulation tools to assist in modeling and evaluation [67]. Node deployment coverage types in 3D WSNs mainly include 3D space and 3D surface. The former primarily utilizes spherical sensing models to achieve complete coverage in regular geometric spaces like cubes, while the latter relies on actual or simulated terrain data for node deployment in surface scenarios. This section focuses on the latter, discussing the construction and representation of 3D surface maps.

2.6.1. Map Sources

Sources of 3D maps primarily fall into three categories: **(1)** real-world map acquisition using sensors or remote sensing technologies; **(2)** function-based or software-assisted generation via simulation tools or terrain modeling algorithms; and **(3)** direct download from public datasets as illustrated in Figure 9.

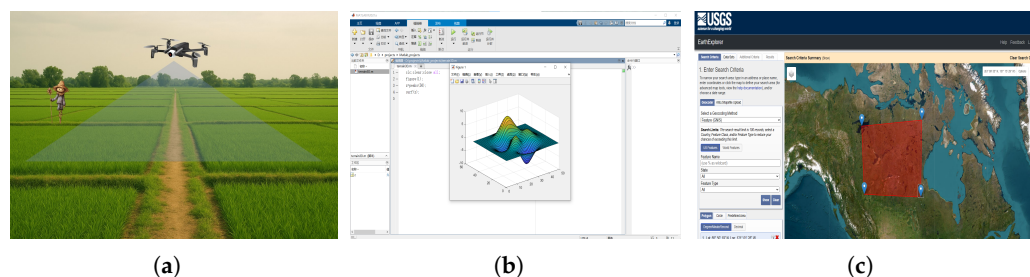


Figure 9. Map sources. (a) Real-world map acquisition. (b) Function generation. (c) Public dataset.

Real-World Map Acquisition: Real-world maps can be constructed through various methods, such as unmanned aerial vehicle (UAV) image capture, point cloud reconstruction, simultaneous localization and mapping (SLAM), and manual annotation. Common devices for map collection include UAVs, LiDAR sensors, RGB-D cameras, GPS modules, and multiview camera systems [68,69]. Although manual annotation is time-consuming, it provides precise and controllable parameters for small-scale scenarios. For example, the authors in [70] manually position buildings and terrain elevations to construct a campus environment and a flat residential area; researchers in [50] model a museum exhibition hall in Zhejiang, with sensors restricted to predefined installation panels; and in [48], satellite imagery is used to sample points every 10 m on Dagong Island, reconstructing a realistic 3D terrain model.

Synthetic Map Generation: Beyond self-constructed maps, some studies employ mathematical functions or modeling software to generate synthetic maps, enabling the rapid creation of terrain environments. For instance, the authors in [49,71] utilize MATLAB's “peak” function to simulate hilly terrains; the study in [21] generates a concave surface using an inverted conical function combined with an Archimedean spiral; in [72], Terragen and

Ridged Perlin Noise are applied to create mountainous terrain, with parameters controlling slope steepness; in [73], the Blender graphic package is used to define plains, plateaus, and hills with custom boundaries; and the authors in [37] present a method for generating random 3D maps using parameterized terrain generation tools available online.

Online Data Acquisition: To avoid overly regular terrains produced by mathematical functions, some studies directly use elevation data from online geographic databases for terrain reconstruction. For example, the authors in [74] use real elevation data from the Geospatial Data Cloud to construct simulation maps; in [75], Zonums Terrain Extractor is used to extract elevation grids from selected regions in Google Maps, constructing four typical terrain scenarios; and in [76], NASA's 2005 SRTM data is used to simulate both urban and forested valley regions near UC Santa Cruz.

2.6.2. Representation Formats of Maps

Once a 3D map is constructed, it must be represented using appropriate data formats to facilitate spatial analysis, sensor deployment, and simulation. Common representation formats include DEM and function-based surfaces.

Representation formats of DEM data include (a) contour-based structures; (b) gridded DEM (GDEM); and (c) triangulated irregular networks (TIN) [77] as shown in Figure 10. Among these, GDEM and TIN are commonly used in 3D WSN deployment.

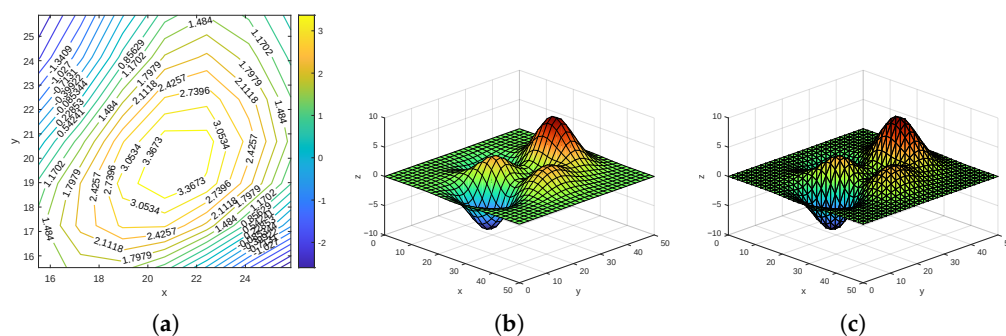


Figure 10. Representation formats for DEM data. (a) Contour model. (b) Grid model. (c) TIN model.

GDEM represents terrain as a regular grid of elevation values stored in a 2D array, with each cell indicating surface height. The accuracy of a GDEM directly depends on its sampling density. In [74], different sampling resolutions are compared to demonstrate the balance between accuracy and simulation performance. Studies in [46,78] also adopt GDEM for map modeling.

TIN constructs irregular meshes using Delaunay triangulation, effectively adapting to a complex terrain. It adjusts point density based on terrain variation, denser in rugged areas and sparser in flat zones, enabling the expression of terrain details like ridges and valleys. TIN-based representation is adopted in studies such as [32,72,73] to model complex terrain features for deployment analysis. Additionally, in [75], the authors introduce the concept of visibility triangles to determine the LoS relationship between adjacent faces and merge visibility triangles to simplify the terrain with TIN. Two triangles are considered mutually visible only if they meet two key requirements: they must share a common edge, and all points in the triangles must have LoS with each other as determined by the Bresenham algorithm. As illustrated in Figure 11, triangles t_1 and t_2 are mutually invisible, while t_3 and t_4 are visible to each other.

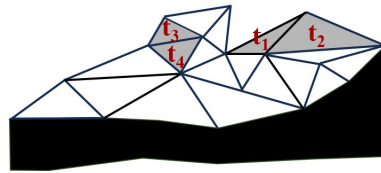


Figure 11. An example of visibility triangles (t_1 and t_2 are invisible, t_3 and t_4 are visible).

Function-based surface representations describe terrain features using continuous mathematical expressions, offering a high degree of flexibility and analytical simplicity. These models are particularly useful for simulating synthetic terrains where parameters such as elevation, slope, and curvature can be precisely controlled. For example, the study in [20] defines the following function as Equation (21) to generate continuous rolling surfaces:

$$z = 80 \cdot \exp\left(-\frac{(x-50)^2 + (y-50)^2}{600}\right) \quad (21)$$

Such mathematical models are especially suitable for controlled environments or algorithm testing scenarios where the terrain structure must be explicitly defined. Similar approaches have also been employed in [79] to construct simulation terrains for node deployment.

3. Mathematical Algorithms for Deployment in 3D WSNs

In recent years, various algorithms have been proposed to optimize node deployment in 3D WSNs. These algorithms can be classified into six categories: classical algorithms, computational geometry algorithms, virtual force algorithms, evolutionary algorithms, swarm intelligence algorithms, and approximation algorithms as shown in Figure 12. This section briefly explains the basic principles and computational models.

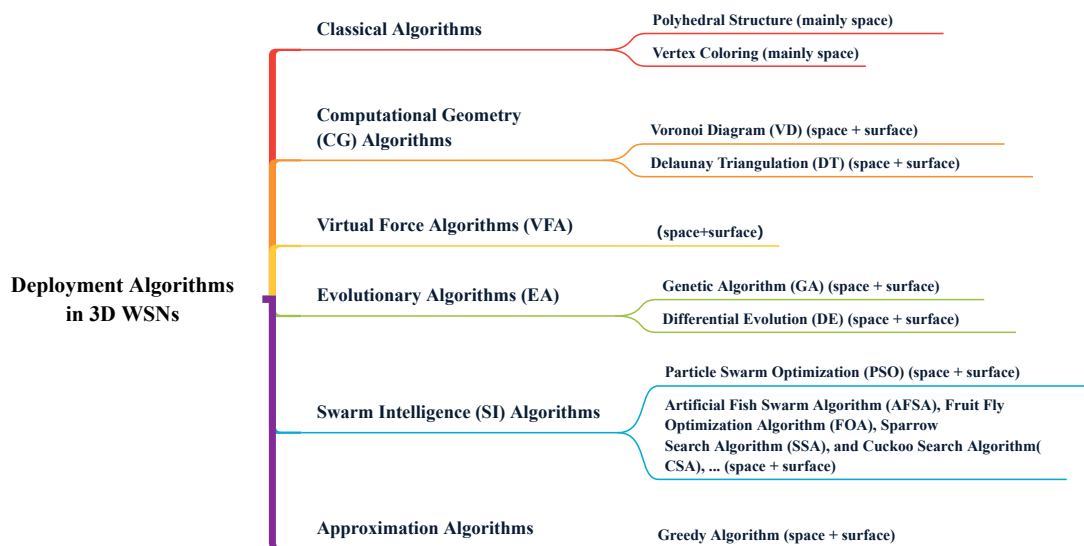


Figure 12. Deployment algorithms in 3D WSNs.

3.1. The Principle of Classical Algorithms

Classical deployment algorithms in 3D WSNs typically rely on geometric partitioning and graph-theoretical methods. These approaches ensure coverage and connectivity through structured spatial filling and discrete node placement. This section discusses two commonly used classical approaches: polyhedral filling algorithms [80] and vertex-coloring-based deployment strategies [81].

3.1.1. Polyhedral Structure

The fundamental idea of the polyhedral structure approach is to treat the 3D monitoring region as a space composed of geometric units, with sensor nodes placed at key points within each unit. This enables regular and uniform node coverage across the entire region. Common filling structures include polyhedrons such as cubes, triangular prisms, quadrangular pyramids, and hexagonal prisms [80] as shown in Figure 13. These structures can either fully fill the 3D space or approximate it.

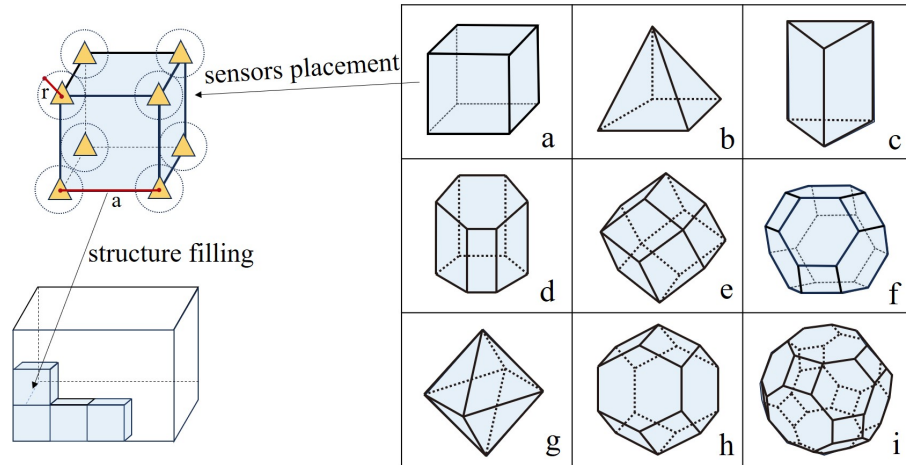


Figure 13. Polyhedral structures and an example of cube-based filling method: (a) cube, (b) quadrangular pyramid, (c) triangular prism, (d) hexagonal prism, (e) rhombic dodecahedron, (f) truncated octahedron, (g) goldberg's equilateral octahedron, (h) elongated dodecahedron, (i) rhombic triacontahedron.

Figure 13 also shows an example of the cube-based filling method. In this case, the target cubic region is divided into smaller cubes, with sensor nodes placed at the vertices of these small cubes. Each node has a spherical sensing range with radius r . By constraining the relationship between the sensing radius r and the cube edge length a , complete coverage of the target space can be theoretically guaranteed [82].

3.1.2. Vertex Coloring

The vertex coloring problem (VCP) aims to assign colors to vertices in a graph such that adjacent vertices have distinct colors, while minimizing the total number of colors used. In sensor deployment, this approach enables coverage of all target points with the minimum number of sensor nodes and ensures no target is redundantly monitored by multiple sensors. To apply VCP, an undirected graph $G = (V, E)$ is constructed, where the vertex set $V(G) = \{v_i\}_{i=1}^m$ represents target points. Edges are defined based on the sensing range R_s and the distance $dist(v_i, v_j)$ between two target points: if $dist(v_i, v_j) > R_s$, an edge $(v_i, v_j) \in E$ exists. This indicates that the two targets cannot be effectively covered by the same sensor and must be monitored by different sensors, represented by adjacent vertices in the graph [81].

To illustrate the process of vertex coloring, Figure 14 presents a simple example. Target points are distributed within a monitoring area, and an undirected graph is built based on their mutual distances. An edge between two vertices means the corresponding target points cannot be covered by the same sensor. After vertex coloring, target points with the same color can be covered by a single sensor, while those with different colors require separate sensors. In this example, three colors are used, indicating at least three sensors are needed to cover all target points.

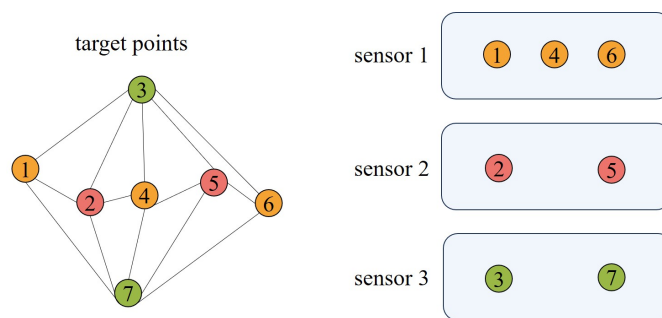


Figure 14. An example of the vertex coloring method.

3.2. The Principle of Computational Geometry Algorithms

Computational geometry algorithms are important for node deployment in WSNs. Among them, the Voronoi diagram (VD) and Delaunay triangulation, as a pair of dual geometric structures, are widely used to describe the spatial relationships between sensor nodes. They can optimize key deployment requirements such as coverage, network connectivity, and energy efficiency.

3.2.1. Voronoi Diagram

The Voronoi diagram [83] is a classical spatial partitioning method that divides a plane into adjacent polygons called Voronoi cells. Each cell is formed by the perpendicular bisectors between the positions of neighboring nodes. In WSNs, the VD partitions the monitoring area into multiple cells, each corresponding to a sensor node. The Voronoi cell contains all points closest to its corresponding node. By defining each node's coverage area, the VD enables accurate evaluation of network coverage and identification of coverage holes. It can also determine the optimal sensing radius, and support energy balancing and self-organization in the network.

3.2.2. Delaunay Triangulation

Delaunay triangulation is the dual structure of the VD, providing an optimal method for partitioning a set of points in a 2D plane into triangles. It ensures that no point lies inside the circumcircle of any triangle, avoiding slender triangles and creating an optimal connection structure. The VD and DT have a clear geometric duality: in WSNs, the VD can be used to optimize coverage performance and detect coverage holes, while the DT can be used to construct communication topologies to ensure connectivity.

Figure 15 shows a VD and its dual DT in a 2D plane, where the black points are sensor nodes. The Voronoi cells partition the target area, and the Delaunay edges represent optimal communication links. This combined representation simultaneously optimizes coverage and network connectivity, and these fundamental principles can naturally extend to 3D environments [73,84].

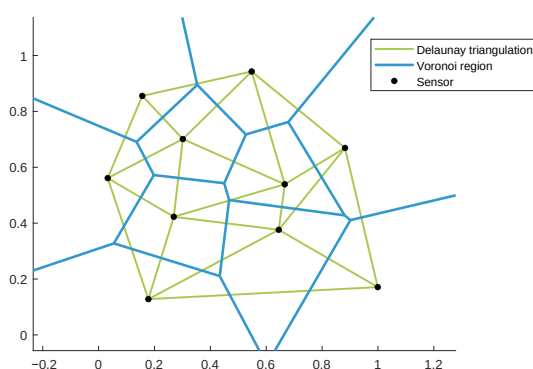


Figure 15. Voronoi diagram and Delaunay triangulation in 2D plane.

3.3. The Principle of Virtual Force Algorithm

The virtual force (VF) algorithm [85] optimizes the deployment of sensor nodes by assuming the existence of virtual forces between nodes, and between nodes and the environment. This approach requires nodes to be homogeneous, mobile, and equipped with positioning capabilities (e.g., via GPS or other localization methods).

In the classical virtual force model, each sensor node s_i is subjected to three types of forces simultaneously: (1) repulsive force $\mathbf{F}_i^R$, which maintains a safe distance between nodes and environmental obstacles; (2) attractive force $\mathbf{F}_i^A$, which guides nodes toward high-priority monitoring areas; and (3) interaction forces $\mathbf{F}_{ij}$ from other nodes s_j , which can be either attractive or repulsive depending on the relative distance and direction between the nodes. A threshold distance d_{th} is set to control the minimum distance between nodes, preventing excessive proximity or separation.

The composite force $\mathbf{F}_i$ acting on node s_i is the vector sum of these forces:

$$\mathbf{F}_i = \mathbf{F}_i^R + \mathbf{F}_i^A + \sum_{\substack{j=1 \\ j \neq i}}^k \mathbf{F}_{ij} \quad (22)$$

Nodes move along the direction of $\mathbf{F}_i$ to approach the system's mechanical equilibrium. Specifically, when the distance between two nodes is less than d_{th} , $\mathbf{F}_{ij}$ acts as a repulsive force to avoid overlap; when greater than d_{th} , it acts as an attractive force to enhance network connectivity. Figure 16 illustrates the interaction forces between nodes at different distances. Assuming $d_{th} = 2 \times R_s$, there is no interaction force between sensors s_4 and s_1 because $d_{14} = d_{th}$; an attractive force $\mathbf{f}_{12}$ acts on s_1 since $d_{12} > d_{th}$; and a repulsive force $\mathbf{f}_{13}$ acts on s_1 when $d_{13} < d_{th}$.

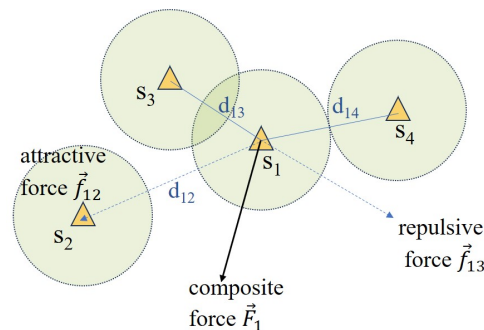


Figure 16. An example of virtual forces on a node.

Additionally, repulsive forces from obstacles or deployment region boundaries can be added to simulate obstruction when nodes approach them [39]. Some studies introduce gravity as an attractive force [20] or targeted attraction from key regions [86], treating these critical areas as “mass points” with gravitational pull to encourage node movement toward high-priority zones. By continuously summing these forces and iteratively updating node positions, the virtual force model effectively enables adaptive sensor deployment in WSNs.

3.4. The Principle of Evolutionary Algorithms

Evolutionary algorithms are a class of optimization methods inspired by natural selection and population evolution mechanisms. They are particularly suitable for handling complex multi-objective problems, as they can explore multiple solutions simultaneously and adapt well to different conditions. The basic process involves continuously evolving the population through initialization, fitness evaluation, selection, and variation operations

to generate new solutions. This section focuses on two representative algorithms: genetic algorithms and differential evolution algorithms.

3.4.1. Genetic Algorithm

The genetic algorithm [87] simulates natural evolution processes. It evaluates the fitness of each individual in the population, selects high-quality parents for crossover and mutation to generate new offspring, and repeats these operations to continuously optimize the population, gradually approaching the optimal solution. Different versions of GA mainly differ in the methods used to select parents and generate new individuals.

GA represents potential solutions using chromosome structures, which are commonly encoded in binary, real numbers, or permutations. Its basic process includes the following steps: **(1) Initial population:** GA starts with a randomly generated population, often initialized using a Gaussian distribution. Each individual is represented by a chromosome composed of a set of variables (genes). The initial population should broadly cover the search space to enhance the algorithm's exploration capability and increase the likelihood of finding the optimal solution. **(2) Fitness evaluation:** The fitness value of each individual is calculated by evaluating its performance on the target function, with higher values indicating better solutions. **(3) Selection:** Based on fitness values, high-quality individuals are selected as parents for the next generation using methods such as roulette wheel selection or tournament selection. These approaches ensure that better solutions have a higher probability of passing on their genes. **(4) Crossover:** The selected parent chromosomes undergo genetic recombination to produce new offspring. Taking uniform crossover as an example, genes are exchanged between parents, which increases diversity and enhances global search ability. **(5) Mutation:** Genes in the offspring chromosomes are randomly modified with a certain probability to maintain diversity and prevent premature convergence to local optima. **(6) Update and iteration:** The new offspring replace the old population, and the processes of fitness evaluation, selection, crossover, and mutation are repeated until the termination conditions are met, such as reaching the maximum number of generations or achieving a satisfactory fitness level. Figure 17 illustrates an example of crossover in Genetic Algorithm.

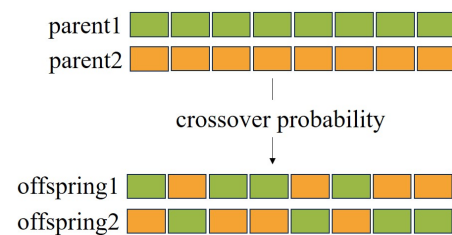


Figure 17. Uniform crossover.

3.4.2. Differential Evolution Algorithm

The differential evolution algorithm [88] is an evolutionary algorithm designed for real-valued optimization problems. Its core lies in using differential information between individuals to guide the search toward the global optimum. It features a unique mutation mechanism and adopts real-valued encoding, avoiding the precision limitations of binary encoding. The specific steps of DE are as follows: **(1) Initial population:** NP D -dimensional real-valued vectors are randomly generated within the parameter space as the initial population. **(2) Mutation:** For each individual, three different individuals $\mathbf{x}_{r1}$, $\mathbf{x}_{r2}$, $\mathbf{x}_{r3}$ are randomly selected to generate a mutation vector $\mathbf{v}_i$, with the amplification of the differential vector controlled by a mutation factor:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (23)$$

where $F \in (0, 1)$ is the mutation factor that controls the amplification of the differential vector. **(3) Crossover:** The mutation vector is combined with the original individual to form a trial vector $\mathbf{u}_i$. A crossover probability and a random dimension ensure that at least one component is derived from the mutation vector:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } \text{rand}(0, 1) < CR \text{ or } j = j_{\text{rand}} \\ x_{i,j}, & \text{otherwise} \end{cases} \quad (24)$$

where $u_{i,j}$ represents the j -th dimension value of the trial vector $\mathbf{u}_i$, $v_{i,j}$ is the corresponding value from the mutation vector $\mathbf{v}_i$, and $x_{i,j}$ is the corresponding value from the original individual $\mathbf{x}_i$. The crossover probability $CR \in [0, 1]$ controls the likelihood of selecting values from the mutation vector, while $j_{\text{rand}} \in [1, D]$ ensures that $\mathbf{u}_i$ inherits at least one component from $\mathbf{v}_i$. **(4) Selection:** The fitness of the trial vector $f(\mathbf{u}_i)$ and the fitness of original individual $f(\mathbf{x}_i)$ is compared, and the better one is retained for the next generation:

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{u}_i, & \text{if } f(\mathbf{u}_i) < f(\mathbf{x}_i) \\ \mathbf{x}_i, & \text{otherwise} \end{cases} \quad (25)$$

where $\mathbf{x}_i^{t+1}$ denotes the solution vector of the i -th individual in the $(t + 1)$ -th generation.

As two representative evolutionary methods, GA and DE are well-suited for solving complex optimization problems in 3D WSN deployment, particularly multi-objective issues involving coverage, connectivity, and energy consumption.

3.5. The Principle of Swarm Intelligence Algorithms

Swarm intelligence algorithms are a class of optimization methods inspired by natural collective behaviors such as bird flocking and fish schooling. These groups search for food through cooperative strategies, where each member adjusts its search direction based on its own experience and others. This section mainly introduces the principle of particle swarm optimization algorithm.

Particle swarm optimization algorithm [89] is a representative swarm intelligence algorithm, inspired by the collaborative behavior of bird flocks in food searching. This method involves a group of particles (candidate solutions) that move through the search space. Each particle adjusts its velocity and position based on its own experience and the global knowledge of the swarm to search for the optimal solution. The basic steps of PSO are as follows: **(1) Initialization:** Each candidate solution is called a “particle”, which represents a point in the search space. For a problem with D variables, the position of the i -th particle is denoted by the vector $\mathbf{x}_i$:

$$\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD}] \quad (26)$$

The entire swarm consists N particles, denoted as

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \quad (27)$$

(2) Velocity Update:

The velocity of each particle in every dimension is updated using three components: inertia, individual cognition, and social cooperation. The update formula is

$$\mathbf{v}_i^{t+1} = \omega \cdot \mathbf{v}_i^t + c_1 \cdot r_1 \cdot (\mathbf{p}_i - \mathbf{x}_i^t) + c_2 \cdot r_2 \cdot (\mathbf{g} - \mathbf{x}_i^t) \quad (28)$$

where ω is the inertia weight, c_1 and c_2 are the acceleration coefficients for the individual and social components respectively, r_1, r_2 are random values generated from a uniform distribution in $[0, 1]$, $\mathbf{p}_i$ is the best position found by particle i , and $\mathbf{g}$ is the best position found by the entire swarm. **(3) Position Update:** Each particle updates its position based on the new velocity:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad (29)$$

(4) Fitness Evaluation and Iteration: Each particle's fitness is evaluated using the objective function and personal and global best positions are updated accordingly. This process repeats until the termination condition is met such as reaching a maximum number of iterations or achieving a desired fitness level.

In addressing node deployment problems in 3D WSNs, PSO and other swarm intelligence algorithms such as artificial fish swarm algorithm [90], fruit fly optimization algorithm [91], sparrow search algorithm [59], and cuckoo search algorithm [92] are widely applied to optimize node placement globally. These algorithms offer flexible and efficient solutions for solving complex deployment tasks in dynamic and constrained environments.

3.6. The Principle of Approximation Algorithms

Calculating exact optimal solutions is often challenging, so approximation algorithms are designed to find near-optimal solutions. These methods can obtain acceptable results within polynomial time, making them widely applicable. Among them, the greedy algorithm is one of the most popular approximation methods, thanks to its simplicity in implementation, theoretical guarantees, and good practical performance. The authors in [93] reviewed the performance guarantees of greedy strategies in solving submodular function-constrained optimization problems. This section focuses on the basic principles of the greedy algorithm. The greedy algorithm [94] constructs solutions step by step through local optimal choices. At each step, it selects the currently "best" option (i.e., a local optimal choice) with the aim of ultimately reaching a solution close to the global optimum. With typically low time complexity, the greedy algorithm is well-suited for resource-constrained WSNs environments.

Taking the target coverage problem as an example, its goal is to deploy the minimum number of sensor nodes while ensuring each target point is covered by at least one sensor. This problem can be modeled as a set cover problem, for which the greedy algorithm provides a classic approximation solution. Assume there is a set of sensor nodes $\mathcal{S} = \{s_i\}_{i=1}^m$, a set of target points $\mathcal{T} = \{t_i\}_{i=1}^n$, and each node s_i covers a subset $X(s_i) \subseteq \mathcal{T}$. The final selected set of sensor nodes is denoted as $\mathcal{S}' \subseteq \mathcal{S}$. The pseudo-code of a basic greedy strategy for the target coverage problem is presented in Algorithm 1:

Algorithm 1: Greedy algorithm for coverage.

Input: $\mathcal{T}, \mathcal{S}$

Output: $\mathcal{S}'$

1 Initialize $\mathcal{S}' \leftarrow \emptyset, \mathcal{U} \leftarrow \mathcal{T}$.

2 **while** $\mathcal{U} \neq \emptyset$ **do**

3 Select sensor $s^* \in \mathcal{S} \setminus \mathcal{S}'$ maximizing uncovered coverage:

4 $s^* = \arg \max_{s \in \mathcal{S} \setminus \mathcal{S}'} |X(s) \cap \mathcal{U}|;$

5 $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{s^*\};$

6 $\mathcal{U} \leftarrow \mathcal{U} \setminus X(s^*).$

7 **return** $\mathcal{S}'$

The greedy algorithm is a fundamental approach for addressing NP-hard problems such as coverage issues. For coverage problems in 3D WSNs, it effectively solves key optimization challenges like minimizing the number of nodes or maximizing the effective coverage area. By formulating these tasks as submodular functions, the greedy algorithm can theoretically provide guaranteed approximation solutions through iteratively selecting the local optima.

4. Literature Review of Deployment Algorithms in 3D WSNs

4.1. Classical Algorithms

Classical algorithms for node deployment in 3D WSNs typically rely on regular polyhedral structures and graph theory methods to achieve efficient node placement. The focus of these methods is to minimize computational costs while ensuring coverage.

In practical urban scenarios, the study in [95] proposes a cuboid-based coverage model. It establishes key mathematical relationships between the sensing radius and coverage area, calculates the minimum number of nodes required to fully cover the target region, and compares the number of nodes needed by different algorithms to achieve the same coverage rate.

Using the binary sensing model, the study in [82] systematically evaluates several geometric structures, including pyramids, triangular prisms, cubes, and hexagonal prisms. Through theoretical analysis, it derives coverage predictions and node requirement estimates. The results show that the pyramid structure achieved the highest coverage but required the most nodes, while the hexagonal prism structure had the lowest coverage but needed the fewest nodes. Based on this, an energy-efficient scheduling algorithm is proposed, which reduced the number of active nodes by approximately half while maintaining high coverage.

The approach in [80] explores a polyhedron-based deployment strategy, using nine distinct convex shapes to fill the target region. The sensor closest to the centroid of each polyhedron is activated to achieve coverage, and the condition $R_s \geq 1.9046R_c$ between the sensing radius R_s and communication radius R_c is restricted to ensure network connectivity. Experimental evaluation demonstrates that the great rhombic dodecahedron provides the best coverage quality among all polyhedra. Although all the above studies employ polyhedral structures for node deployment, their sensor placement strategies differ. In [82], sensors are placed at the vertices of the polyhedra; in [80], sensors are placed at the centroid of each polyhedron; and the study in [95] adopts a hybrid placement strategy, deploying sensors at the eight vertices and the centroid of the cuboid.

Some studies have innovatively applied traditional polyhedral tiling techniques to deployment problems on 3D surfaces. In [96], an approximate surface model is constructed from collected sample points. The entire 3D space is then filled with truncated octahedra, and the centers of those octahedra intersecting the surface are projected onto their nearest surface points, which serves as the final deployment locations of the sensors.

Graph coloring is also a classical deployment algorithm. In [53], the authors propose a deterministic deployment algorithm based on vertex coloring to address the K -coverage problem in 3D space. This method determines the required number of sensors based on the spatial coordinates of targets, derives optimal sensor positions, and introduces a breadth-first search algorithm for relay node deployment to enhance network connectivity.

A similar method is presented in [57], where the authors design a deployment algorithm based on the vertex coloring method to determine the number and optimal placement of sensors, achieving 100% target coverage in 3D space. For connectivity assessment, a breadth-first search algorithm is used to count the number of sensors maintaining active connections to the base station.

To conclude this section, a comparative summary of classical deployment algorithms in 3D WSNs is presented in Table 5, highlighting the coverage type, deployment strategy, adopted sensing model, optimization objectives, connectivity considerations, and map sources.

Table 5. Comparison of classical algorithms in 3D WSN deployment.

Coverage Type	Algorithm	Strategy	Sensing Model	Objective(s)	Connectivity	Map Source
Surface	[96]	truncated octahedron-based	binary	• maximize area coverage • minimize no. of nodes • minimize overlapping area	N/A	real-world
Space	[53]	vertex coloring-based	binary	• minimize no. of nodes	full connectivity	cube
	[57]	vertex coloring-based	binary	• minimize no. of nodes	full connectivity	cube
	[80]	9 convex polyhedrons-based	binary	• minimize no. of nodes	full connectivity	cube
	[82]	prism, cube, pyramid, hexagonal prism-based	binary	• maximize area coverage • minimize no. of nodes • minimize energy consuming	N/A	cube
	[95]	cuboid-based	probabilistic	• minimize no. of nodes	N/A	cube

4.2. Computational Geometry Algorithms

Computational geometry algorithms utilize spatial partitioning techniques such as VD and DT to precisely deploy sensors, and they are particularly effective in improving coverage and connectivity in complex 3D WSNs.

In [97], an energy-efficient coverage enhancement method was proposed, which combined 3D Voronoi partitioning with the k -means algorithm. The monitoring space was first divided into Voronoi cells, and then nodes were clustered using an energy-aware k -means algorithm. Each cell was covered by a high-energy dominant node, while other nodes remained in sleep mode. When the energy of the dominant node drops below a threshold, a candidate node is activated to replace it. This strategy ensures full coverage while reducing the number of active nodes, thereby lowering energy consumption and extending network lifetime.

The study in [98] improves the traditional VD method by introducing two geometric structures, i.e., cubes and regular dodecahedrons, to enhance network coverage. The algorithm organizes sensor nodes into cooperative structural units and dynamically adjusts their positions. Compared with traditional VD methods, this approach achieves better coverage performance while reducing computational complexity and deployment time.

For underwater WSNs, the authors in [13,35] adopt depth adjustment schemes based on Voronoi diagrams to solve node deployment problems. Both studies use a layered deployment strategy, initially deploying nodes randomly on the water surface to form a connected network. In [13], it assumes that the communication radius is much larger than the sensing radius to ensure connectivity, and each node moves only once. In [35], connectivity is maintained by keeping the distance between a sinking node and its upper-layer neighbors within the communication radius R_c . Additionally, a periodic initialization process is introduced, allowing nodes to float to the surface and redeploy periodically to reduce horizontal drift caused by water currents.

The deployment method in [73] combines Delaunay tetrahedralization and 3D Voronoi diagrams. It first uses the Delaunay algorithm to divide the surface into tetrahedrons, with the centers of these tetrahedrons serving as potential sensor positions. The Dijkstra algorithm is used to check network connectivity, and overlapping nodes are dynamically eliminated to optimize coverage and connectivity. This method is evaluated on flat, plateau, and hilly terrains, showing robust performance in various 3D terrains.

Focusing on 3D surfaces, the study in [84] proposes a deployment strategy that combines Voronoi diagrams and Delaunay triangulation. The space is first divided into Voronoi sectors, and then Delaunay triangulation is used to ensure network connectivity. Sensor nodes are placed either at the centers of Voronoi cells or the vertices of Delaunay triangles. Since Delaunay triangulation inherently maintains connectivity, this method eliminates the need for additional validation, simplifying the deployment process.

In [32], mobile sensors with an adjustable and non-uniform sensing radius are used to enhance network flexibility. The deployment algorithm minimizes the maximum sensing radius while achieving K -coverage to reduce energy consumption (as energy usage increases with sensing radius). By iteratively computing K -order Voronoi cells, positioning sensors at the Chebyshev centers of these cells, and adjusting sensing ranges accordingly, the network's energy consumption is effectively balanced.

To conclude this section, Table 6 compares computational geometry-based deployment algorithms in 3D WSNs, focusing on the coverage type, computational geometry structures used, sensing model, node type, optimization objectives, control architecture (centralized or distributed), and map sources.

Table 6. Comparison of CG algorithms in 3D WSN deployment.

Coverage Type	Algorithm	CG Structure	Sensing Model	Node Type	Objective(s)	Centralized/Distributed	Map Source
Surface	[32]	VD	binary	mobile heterogeneous	• full area coverage • minimize the maximum sensing radius of nodes	centralized	not state
	[73]	VD + DT	binary	static homogeneous	• maximize area coverage • minimize no. of nodes	centralized	not state
	[84]	VD + DT	binary	static homogeneous	• maximize area coverage	centralized	tool generation
Space	[13]	VD	probabilistic	mobile homogeneous	• maximize area coverage • minimize energy consuming	distributed	cube
	[35]	VD	probabilistic	hybrid homogeneous	• maximize area coverage • maximize network lifetime	distributed	cube
	[97]	VD	probabilistic	mobile heterogeneous	• maximize area coverage • minimize energy consuming	centralized	cube
	[98]	VD	binary	mobile homogeneous	• maximize area coverage • minimize deployment time • minimize move distance	centralized	cube

4.3. Virtual Force Algorithm

Multiple studies have applied virtual force algorithms to node deployment in 3D WSNs, aiming to enhance coverage and connectivity while addressing challenges such as node oscillation and energy consumption.

The studies in [20,99] explore the application of distributed virtual force algorithms in deploying mobile sensors on 3D surfaces to achieve full coverage and maintain connectivity. Specifically, the authors in [20] compare two initialization scenarios: random deployment across the mountain surface and deployment starting from the bottom of the mountain. The algorithm in [99] is validated on multiple sloped 3D surfaces. However, both studies overlook the issue of node oscillation, which is a common drawback in virtual force-based methods.

To tackle the limitations of the aforementioned algorithms, the authors in [79] develop a two-phase deployment strategy that simultaneously optimizes energy efficiency and node oscillation in 3D mountain environments. This strategy enables stable sensor movement across various terrain types, resolving the oscillation problem while maintaining good coverage, with nodes naturally stopping movement once full coverage is achieved.

Another focus is on addressing coverage holes. In [62], the authors aim to solve coverage holes caused by irregular deployment or sensor failures. It adopts a hybrid node strategy combined with virtual force adjustments: first, it detects coverage holes using computational geometry methods, then redeploys redundant mobile sensors. This significantly improves the coverage quality and extends network lifetime.

In 3D space, the study in [86] employs a virtual force algorithm to maximize coverage and maintain connectivity, introducing a density control mechanism to prevent uneven node distribution. However, it has high requirements for each node's precise distance estimation, obstacle detection, and relative positioning capabilities, which may pose challenges in practical applications.

The studies in [39,100] adopt Voronoi diagram-based virtual force strategies to optimize coverage and minimize energy consumption. They divide the 3D space using Voronoi diagrams and adjust node positions via virtual forces, ensuring connectivity by maintaining an appropriate ratio between the communication radius and sensing radius. The researchers in [39] use directional and rotatable sensors, prioritizing rotational adjustments to reduce movement costs; the authors in [100] employ a probabilistic sensing model and restrict the movement of redundant sensors that do not cover any targets.

Unlike the aforementioned distributed virtual force algorithms, the authors in [101] adopt a centralized approach for 3D space coverage. Although it exhibits high performance in experimental environments, the centralized design may limit the scalability of large-scale networks. The authors note that this algorithm can also be applied to 3D surface deployment and directional sensor coverage.

To conclude this section, Table 7 compares virtual force-based deployment algorithms in 3D WSNs, focusing on the coverage type, sources of attractive and repulsive virtual forces, sensing model, control architecture (centralized or distributed), and map sources.

Table 7. Comparison of virtual force (VF) algorithms in 3D WSN deployment.

Coverage Type	Algorithm	Source of Attractive VFs	Source of Repulsive VFs	Sensing Model	Centralized/Distributed	Map Source
Surface	[20]	• sensors • gravity	• sensors	binary	distributed	function
	[62]	• sensors	• sensors	probabilistic	distributed	function + real-world
	[79]	• sensors • gravity	• sensors	binary	distributed	function
	[99]	• sensors	• sensors	binary	distributed	function
Space	[39]	• sensors • target areas	• sensors • RoI boundaries • obstacles	directional	distributed	cube
	[86]	• sensors • particular area	• sensors • obstacles	binary	distributed	cube
	[100]	• sensors • target areas	• sensors • RoI boundaries • obstacles	probabilistic	distributed	cube
	[101]	• sensors	• sensors • RoI boundaries	binary	centralized	cube

4.4. Evolutionary Algorithm

Evolutionary algorithms often adopt multi-objective optimization techniques and hybrid strategies, combining GA with other heuristic algorithms to balance coverage, connectivity, energy consumption, and localization accuracy in complex environments.

In [46], the authors introduce a genetic algorithm with a wavelet-guided mutation operator. By leveraging the signal processing capabilities of wavelet transform, this approach significantly improves both coverage performance and algorithm convergence speed.

Multi-objective genetic algorithms extend the basic genetic algorithm by optimizing multiple conflicting objectives simultaneously. The study in [14] proposes different optimization objectives for military and civilian scenarios. Through adaptive fitness functions and constraint adjustments, it solves the deployment problems in both scenarios.

The authors in [47] combine an adaptive guided genetic operator with multi-objective optimization for 3D surface deployment. This method considers forbidden areas where sensors cannot be placed and introduce two optimization techniques to enhance solution quality and computational performance. In [102], a three-phase deployment scheme is proposed, along with a multi-response k -means GA. This approach optimizes five parameters simultaneously and applies Pareto optimality to balance multiple objectives.

To address the limitations of traditional genetic algorithms, the authors in [33] combine an improved particle swarm optimization algorithm with genetic algorithms, improving the convergence speed, network coverage, and energy efficiency. The study in [51] combines the non-dominated sorting genetic algorithm II with the ant colony optimization algorithm, further considering network connectivity and the localization accuracy of mobile nodes, confirming that the combined algorithm can significantly improve performance.

To tackle the localization accuracy issue raised in [51], the authors in [52] apply the DV-Hop algorithm and a localization correction model based on the received signal strength indicator and frame erasure rate, improving coverage and localization quality, but without considering other objectives such as connectivity or network lifetime.

DE is also commonly used to solve deployment problems in 3D WSNs. In [103], the 3D indoor deployment problem is formalized as an eight-objective optimization task, and an improved multi-objective differential evolution algorithm is adopted, tested in buildings with dynamic features like walls, doors, and windows.

Authors in several studies, such as [41,55,70], employ distributed multi-objective differential evolution algorithms to optimize the deployment of 3D WSNs.

Specifically, the authors in [55] propose an improved differential evolution algorithm based on crossover rate ranking and polynomial mutation, dividing nodes into data sensor nodes and cluster head nodes, and considering objective functions such as coverage, lifetime, connectivity, and reliability. The clustering method used in the previous text is shown in Figure 18. In addition, security management can be integrated into the selection of cluster head nodes by evaluating node trust levels to ensure reliable cluster heads, thus reducing attack risks and improving network security [104].

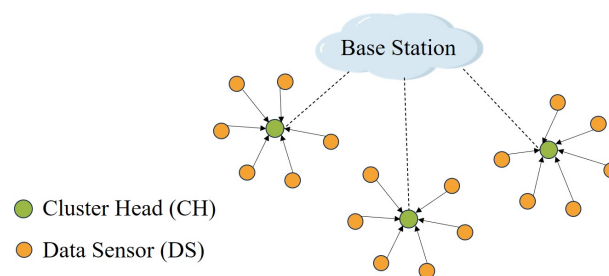


Figure 18. The clustering method in WSNs.

In [70], a new distributed parallel differential evolution algorithm is introduced for deploying heterogeneous and directional sensors on smart city surfaces. Using 3D city maps, the deployment problem is constructed as a multi-objective optimization problem, meeting key requirements for coverage, connectivity, and reliability.

Unlike [55,70], where the authors focus on 3D surfaces, the authors in [41] address the coverage challenges inside 3D ship hulls. The proposed solution optimizes three indicators:

coverage, network lifetime, and reliability. Comparative tests show that this method outperforms traditional algorithms in optimization effect and computational efficiency.

To conclude this section, Table 8 compares evolutionary algorithm-based deployment algorithms in 3D WSNs, focusing on coverage objects, optimization strategies, objectives, sensing models, node types, control architectures, and map sources.

Table 8. Comparison of evolution algorithms (EAs) in 3D WSN deployment.

Coverage Type	Algorithm	Strategy	Objectives	Sensing Model	Node Type	Centralized/Distributed	Map Source
Surface	[46]	GA	single	probabilistic	static homogeneous	centralized	synthetic
	[47]	GA	multiple	probabilistic	static homogeneous	centralized	synthetic
	[55]	DE	multiple	directional	static homogeneous	distributed	dataset
	[70]	DE	multiple	directional	static homogeneous	distributed	real-world
Space	[14]	GA	single	probabilistic	static homogeneous	centralized	not state
	[33]	GA	single	binary	static homogeneous	centralized	cube + real-world
	[41]	DE	multiple	directional	static heterogeneous	distributed	real-world
	[51]	GA	multiple	binary	hybrid homogeneous	centralized	not state
	[52]	GA	multiple	binary	hybrid homogeneous	centralized	real-world
	[102]	GA	multiple	binary	static heterogeneous	centralized	not state
	[103]	DE	multiple	binary	hybrid homogeneous	centralized	real-world

4.5. Swarm Intelligence Algorithm

Many studies have adopted swarm intelligence algorithms for node deployment in 3D wireless sensor networks. These methods often utilize hybrid or parallel strategies to overcome local optima and accelerate convergence speed, effectively addressing challenges in complex and dynamic deployment scenarios.

In the field of underwater deployment, the study in [105] considers the impact of water currents and waves on node movement, proposing an improved fruit fly algorithm to help mobile nodes resist such influences. Compared with the improved adaptive particle swarm optimization algorithm, this approach achieves better coverage performance. For underwater wireless sensor networks, the authors in [106] use the cuckoo search algorithm to maximize coverage and minimize the number of nodes. Through random deployment and position optimization, it outperforms random deployment strategies in both large-scale and small-scale networks. Furthermore, the study in [92] further improves the cuckoo search algorithm by introducing “survival of the fittest”, dynamic discovery probability, and rotational adaptive strategies. Based on a directional sensing model, it maximizes coverage while minimizing the required number of nodes.

For 3D surface deployment, the authors in [49] introduce an enhanced black hole algorithm, which incorporates a multi-black hole model, dynamic weight factors, and mutation operations to solve node coverage problems, outperforming the original black hole algorithm in both benchmark tests and 3D coverage simulations. The study in [45] proposes an improved marine predator algorithm, which divides the surface into grid cells and optimizes search strategies and dynamic learning mechanisms to effectively handle sensor deployment challenges on undulating and rugged terrains. The researchers in [71]

address the high memory requirements of existing heuristic algorithms by developing a compact particle swarm optimization algorithm suitable for 3D surface deployment, combined with a Gaussian perturbation strategy to avoid local optima.

While most studies focus on maximizing coverage to improve overall deployment performance of wireless sensor networks, ensuring connectivity is equally important for reliable data transmission and network stability in practical applications. Hence, more research considers both coverage and connectivity. The authors in [31] propose a 3D surface optimization method based on Delaunay triangulation and an enhanced butterfly optimization algorithm, solving the blind area problem on 3D surfaces by constructing a 3D probabilistic coverage model, with experimental results showing excellent performance in both coverage and connectivity. In [44], the authors present an enhanced grey wolf optimization algorithm for 3D surface node deployment, which enhances the exploration capability based on standard methods and performs well in coverage and connectivity in tests on 12 well-known benchmark functions. The study in [59] proposes a coverage optimization strategy for 3D wireless sensor networks based on an improved sparrow search algorithm, introducing a safety threshold attenuation function and a stagnation update mechanism to overcome local optima and slow convergence, with experimental results demonstrating improved coverage efficiency, connectivity, and reduced deployment costs. In [58], a hybrid lion swarm optimization method is introduced for spatial deployment optimization, which is successfully applied to coverage optimization in 3D wireless sensor networks after evaluating 100 standard benchmark functions, showing good performance in both coverage and connectivity. The researchers in [60] propose a novel distance calculation method based on signal attenuation, limiting the maximum node distance through a path loss threshold, and combining an enhanced particle swarm optimization algorithm with simulated annealing to enhance the ability to escape local optima. Compared with traditional particle swarm optimization algorithms and 3D self-deployment algorithms, it performs better in coverage and convergence, and the signal attenuation model is verified through actual deployment. The authors in [18] propose a multi-objective optimization algorithm combining particle swarm optimization and multi-agent systems, addressing five key network performance indicators: coverage, connectivity, localization accuracy, link quality, and network utilization, with effectiveness verified through experiments on an actual deployment platform.

In mobile node deployment, node movement leads to additional energy consumption. Improper management may cause nodes to fail prematurely, affecting network stability and lifetime. Therefore, energy control and maximizing network lifetime have become important goals in 3D wireless sensor network deployment optimization. The study in [107] developed a multi-objective competitive learning optimizer for 3D space sensor coverage optimization, aiming for high coverage and low energy consumption, balancing these goals by generating a Pareto front, and effectively reducing energy consumption and extending network lifetime by reducing movement distance and position fluctuations during deployment. The researchers in [108] focus on maximizing coverage and maintaining connectivity, proposing a combined algorithm based on distributed particle swarm optimization and 3D virtual force algorithms. This method randomly initializes the positions of static and dynamic nodes, adjusts dynamic nodes for 3D space deployment, and evaluates network lifetime by tracking the failure time of sensor nodes.

Most swarm intelligence algorithms have a critical flaw of easily falling into the local optima. Many researchers have proposed various enhancement strategies, among which parallel computing has shown great effectiveness in improving algorithm performance. The authors in [109] propose an improved parallel tunicate swarm algorithm, which divides the population into independent subgroups that exchanged information periodically,

enabling more effective optimization of sensor node deployment under limited conditions to maximize network coverage. In [90], the authors introduce a chaotic parallel artificial fish swarm algorithm to enhance the global search ability of the artificial fish swarm algorithm, adopting an elite selection strategy to avoid local optima and solve underwater coverage problems. The study in [40] proposes a cooperative co-evolution particle swarm optimization algorithm for industrial wireless sensor network deployment with obstacles, aiming to maximize coverage and minimize energy consumption. Test results show that the distributed parallel technology of this method reduces the optimization time and avoids falling into the local optima.

In swarm intelligence research, traditional single algorithms usually rely on specific heuristic rules or optimization strategies. To overcome the limitations of single methods, hybrid swarm intelligence algorithms combine the advantages of multiple algorithms. Recent studies have focused on hybridizing different swarm intelligence technologies to leverage the strengths of each algorithm, especially in complex multi-objective optimization problems. The researchers in [91] combine the fruit fly optimization algorithm and the bat optimization algorithm for sensor node localization, integrating the exploration ability of the former and the exploitation ability of the latter, showing excellent performance in coverage, reliability, and convergence speed but with a longer running time. The authors in [48] combine the evolutionary model of the whale optimization algorithm (WOA) with the strong cooperative evolution ability of the shuffled frog leaping algorithm (SFLA). The WOA mimics the bubble-net hunting strategy of humpback whales, while the SFLA combines local search and global information sharing through a cooperative evolution strategy. This combination achieves effective signal coverage in practical wireless sensor networks.

To conclude this section, Table 9 compares SI-based deployment algorithms in 3D WSNs, focusing on coverage type, adopted swarm intelligence strategies, optimization objectives, sensing model, node type, control architecture, and map sources.

Table 9. Comparison of swarm intelligence (SI) algorithms in 3D WSN deployment.

Coverage Type	Algorithm	Strategy	Objective (s)	Sensing Model	Node Type	Centralized / Distributed	Map Source
Surface	[31]	BOA	single	probabilistic	static	centralized	not state
	[44]	GWO	single	binary	static	centralized	function
	[45]	MPA	single	probabilistic	static	centralized	function
	[48]	SFLA + WOA	single	binary	static	centralized	real-world
	[49]	BHA	single	binary	static	centralized	function
	[71]	PSO	single	binary	static	centralized	function
	[92]	CSA	single	directional	static	centralized	not state
	[109]	PTSA	single	binary	static	distributed	not state
Space	[18]	PSO	multiple	binary	hybrid	distributed	real-world
	[40]	PSO	multiple	directional	static	distributed	real-world
	[58]	LSOA	multiple	binary	static	centralized	cube
	[59]	SSA	single	binary	mobile	centralized	cube
	[60]	PSO	multiple	probabilistic	static	centralized	real-world
	[105]	FOA	single	probabilistic	mobile	centralized	cube
	[106]	CSA	single	binary	static	centralized	cube
	[107]	CLO	multiple	binary	mobile	centralized	cube
	[108]	PSO	multiple	probabilistic	hybrid	distributed	cube
	[90]	AFSA	single	binary	static	distributed	cube
	[91]	FOA + BA	single	binary	static	centralized	cube

4.6. Approximation Algorithm

Multiple studies have focused on approximation algorithms for sensor node deployment in 3D WSNs, aiming to balance coverage optimization and computational efficiency. Greedy algorithms are commonly used to reduce the number of sensors while maintaining coverage and connectivity, often transforming deployment problems into set cover models.

The study in [74] models the surface using real elevation data and deployed nodes using a grid-scanning-based greedy algorithm. Compared with heuristic algorithms such as PSO, GA, and ACO, the results show that this algorithm improves node coverage efficiency in complex terrains, reduces deployment costs, and lowered computational complexity.

In [72], three approximation algorithms for WSN deployment are introduced. One is a greedy algorithm similar to that in [74]; another is a polynomial-time approximation algorithm based on a mobile strategy; and the third combines the greedy method with the mobile strategy. The authors point out that future research could consider dynamic sensors and network connectivity.

The authors in [17] study node deployment in wireless video sensor networks with position constraints in 3D indoor spaces. It combines a greedy algorithm with enhanced depth-first search to achieve full coverage with the minimum number of nodes. The authors indicate that future research directions include network connectivity and K -coverage in obstacle environments. In a similar scenario, the researchers in [50] use a greedy algorithm to minimize the number of sensors, achieving double coverage of the target area while maintaining network connectivity. The results show that this algorithm outperforms GA-based heuristic algorithms and approaches optimal performance in small-scale problems.

To conclude this section, the Table 10 presents a comparison of approximation-based deployment algorithms in 3D WSNs, highlighting coverage type, strategy, sensing model, optimization objectives, connectivity considerations, and map sources.

Table 10. Comparison of approximation deployment algorithms in 3D WSNs.

Coverage Type	Algorithm	Strategy	Sensing Model	Objective(s)	Connectivity	Map Source
Surface	[72]	greedy algorithm + shift strategy-based approximation algorithm	binary	• full 1-coverage • minimize no. of nodes	N/A	tool generation
	[74]	greedy algorithm	probabilistic	• maximize area coverage	N/A	dataset
Space	[17]	greedy algorithm	directional	• full 1-coverage • minimize no. of nodes	N/A	cube
	[50]	greedy algorithm	binary	• full k-coverage • minimize no. of nodes	full connectivity	real-world

5. Comparative Analysis of Deployment Approaches

This section systematically reviews existing deployment algorithms, including classical algorithms, computational geometry (CG) algorithms, virtual force algorithms, evolutionary algorithms, swarm intelligence (SI) algorithms, and approximation algorithms. It analyzes their characteristics in terms of coverage capability, node mobility, system architecture, and multi-objective optimization ability. Table 11 provides a detailed comparison of these algorithms, summarizing their performance and applicability in different deployment scenarios.

Classical algorithms mainly rely on predetermined rules and simple calculations, making them suitable for static node deployment. They often use geometric tiling and graph coloring techniques, with optimization objectives typically limited to coverage and connectivity. However, the lack of dynamic adjustment mechanisms restricts their

performance in complex 3D environments, where they struggle to handle multi-objective trade-offs, thereby limiting their scalability and practicality.

CG algorithms utilize strict geometric relationships for node positioning and coverage optimization, making them suitable for scenarios with a small number of nodes and regular regions, as they can provide uniform coverage distribution. Nevertheless, they have high computational complexity and limited ability to handle multi-objective problems, usually treating coverage as the main objective and connectivity as a secondary constraint.

VF algorithms dynamically adjust node positions by simulating physical forces, showing high efficiency in mobile WSN deployment. They are well-suited for distributed architectures, which can reduce communication costs and enhance single-point failure recovery capabilities.

Figure 19 above illustrates the communication patterns and structural differences between centralized and distributed architectures. However, VF algorithms usually assume uniform sensing and communication ranges, limiting their applicability in heterogeneous WSNs, and they may face issues like oscillation and instability. EA and swarm intelligence algorithms are suitable for multi-objective optimization, as they can simultaneously handle requirements such as coverage, connectivity, and energy consumption, thanks to their strong global search and adaptive capabilities. However, they have high computational costs, slow convergence, and a tendency to fall into local optima. To address these shortcomings, hybrid models and adaptive parameters are often introduced to improve stability and efficiency.

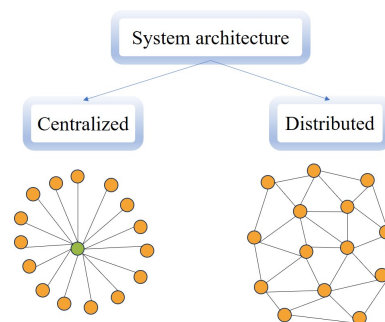


Figure 19. Centralized vs. distributed architectures in WSNs.

Approximation algorithms have high computational efficiency, making them suitable for rapid deployment and large-scale optimization tasks. But they are typically designed for one or two objectives, lacking support for multi-objective scenarios, so they are often combined with other optimization strategies in such cases.

Table 11. Comparison of 3D WSN deployment algorithms.

Algorithm	Coverage	Node Mobility	Centralized/ Distributed	Advantages	Limitations
Classical Algorithm	mainly space	static	centralized	• simple computation • clear deployment rules	• low flexibility • poor terrain adaptability
CG	both	both	both	• clear geometric relationships • uniform coverage	• high computational complexity
VF	both	mobile	both	• strong adaptability • dynamic adjustment capability	• prone to node oscillation
EA	both	both	both	• powerful global optimization	• high resource cost
SI	both	both	both	• strong adaptability • few parameters	• easily falls into local optima
Approximation Algorithm	both	static	centralized	• high computational efficiency	• limited solution quality

6. Case Study

To better illustrate how the reviewed deployment strategies can be applied in practice, this survey concludes with a case study. The case study not only demonstrates the practical relevance of theoretical deployment methods but also highlights how deployment algorithms can be adapted to address real-world challenges in agricultural environments. By focusing on the deployment of solar insecticidal lamps (SILs) in 3D agricultural environments, it provides a concrete example that bridges the gap between conceptual research and field applications.

With the development of smart agriculture, there is an increasing demand for automation and precision in pest control. SILs, as an environmentally friendly technology, are widely used in agricultural environments such as farmland, orchards, and greenhouses. They can achieve real-time pest trapping and have become an important part of modern pest monitoring and management systems.

In 2D environments, the deployment of SILs is typically based on the planar assumption, where both crop distribution and device placement are constrained to a horizontal surface. These 2D areas can be either regular or irregular, and some may contain obstacles. Node locations can also be restricted; for instance, in some studies, installations are limited to field ridges. Coverage types are classified into area coverage, point coverage, and barrier coverage. Common deployment methods in 2D WSNs include grid-based methods, approximation algorithms, and heuristic algorithms, which aim to ensure area coverage while maintaining connectivity or reducing costs. While these methods perform well in practical 2D applications, they show limited adaptability in complex 3D terrains. The construction of a 2D map can be achieved through manual annotation or the download of public datasets. For example, the authors in [110] combine aerial imagery analysis with computational geometry techniques to generate accurate agricultural maps as shown in Figure 20.

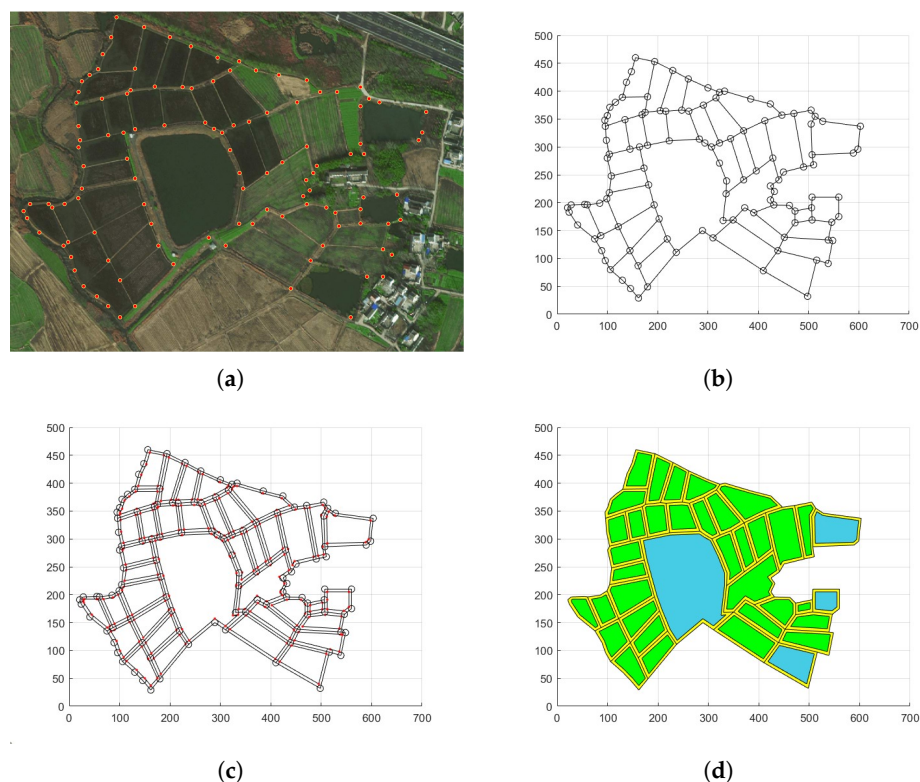


Figure 20. A method for constructing a 2D map. (a) Vertices marked by manual. (b) Minimum circles determined manually. (c) Subcircles determined by the offset algorithm. (d) AnaMap that simulated the actual farmland. Source: [110].

Currently, a number of studies have focused on the deployment of SILs in 2D WSNs. Approximation algorithms are frequently employed in real-world SIL deployments due to their simplicity and computational efficiency. For instance, the authors in [111] propose a multi-objective method based on GA to balance deployment efficiency and cost, achieving full coverage with about 11 nodes under a given sensing radius, whereas other methods typically require 14–28 nodes. In [112], the authors consider both coverage and connectivity, aiming to maximize network weight while minimizing deployment cost; in this study, SILs are deployed in irregular farmland areas to enhance the overall benefit. The study in [113] also targets full coverage and connectivity, with the additional objective of maximizing the overlap rate to reinforce network reliability, while reducing the average number of nodes by about 13.5% compared to other methods. The authors in [114] focus on weight maximization and cost reduction while maintaining connectivity in WSNs to support network performance, improving coverage uniformity by approximately 45–50%. In [115], the authors embed SILs into UAVs to create a mobile network system, utilizing a hybrid method that combines convex hulls and minimum spanning trees (MST) to form closed barrier coverage against pest migration, while reducing the required number of UAVs by 10–30% compared to other deployment methods. Moreover, the authors in [116] adopt a probabilistic sensing model to enable full-area coverage in irregular real-world farmland while maintaining connectivity. In [117], an approximation-based approach is employed to maximize coverage under a limited budget, achieving a 2–30% higher coverage rate compared to other methods. The authors in [118] explore the use of SILs installed with directional and adjustable camera nodes to monitor system status and detect potential intrusions, while reducing the total cost by approximately 15–20% without compromising monitoring performance. In [119], the authors also use SIL nodes with adjustable camera nodes. They introduce a bi-level algorithm to optimize the deployment of SILs nodes and cameras, ensuring area and target coverage, while reducing the total cost by 12.5% and 7.8% compared to two alternative algorithms. The authors in [120] focus on partial coverage and energy harvesting for SILs deployment. They propose an approximation algorithm to achieve high energy harvesting potential, network connectivity, and ensure coverage of at least 80%, while reducing the number of nodes by 6–34%. The above studies are summarized in Table 12.

As agricultural scenarios evolve from flat 2D planes to complex 3D terrains—including terraced fields, hilly landscapes, and vertical cultivation systems—the difficulty of deploying SILs has increased significantly. A comparison of SIL deployment in 2D versus 3D environments is presented in Figure 21.



Figure 21. Comparison of 2D and 3D SILs deployment: (a) 2D SILs deployment, (b) 3D SILs deployment.

Deploying SILs in 3D agricultural environments faces multiple challenges. Firstly, the irregular terrains of terraced fields and hilly landscapes create non-uniform deployment surfaces, requiring the careful consideration of elevation changes and physical accessibil-

ity constraints. Secondly, occlusions caused by complex terrains may lead to coverage blind areas, which must be identified and addressed. Additionally, some regions may lack road accessibility or mechanical operability, limiting the feasibility of physical installation and further reducing the number of valid deployment locations. Meanwhile, pest infestations often exhibit an uneven spatial distribution, with outbreaks concentrated in specific “hotspots”. Thus, deployment strategies must adapt to the spatial distribution of pest density, enhancing coverage in high-risk areas. Collectively, these factors impose strict requirements on system models and algorithm design.

Table 12. Comparison of case studies on 2D SIL deployment.

Case	Year	Sensing Model	Node Mobility	Algorithm	Covered Type	Objective(s)	Target Area	Node Loc. Cons.	Obstacle	Map Source
[111]	2020	binary	static	GA	area	• full coverage • maximize overlap rate • minimize no. of nodes	irregular	✓	✓	real-world
[112]	2020	binary	static	AA	area	• full coverage • full connectivity • maximize the weight • minimize costs	irregular	✓	✓	real-world
[113]	2021	binary	static	AA	area	• full coverage • full connectivity • maximize overlap rate	irregular	✓	✓	real-world
[114]	2022	binary	static	AA	area	• full coverage • full connectivity • maximize the weight • minimize costs	regular	✓	✓	real-world
[115]	2022	probabilistic	mobile	CG + MST	barrier	• closed barrier coverage • minimize no. of nodes	regular	×	×	author-defined
[116]	2023	probabilistic	static	AA	area	• full coverage • full connectivity • minimize no. of nodes	irregular	✓	✓	real-world
[117]	2024	binary	static	AA	area	• maximize coverage • within a limited budget	regular	✓	×	real-world
[118]	2025	binary + directional	static	AA	area	• full coverage • minimize no. of nodes	irregular	×	×	author-defined
[119]	2025	binary + directional	static	AA	area + point	• full coverage • minimize costs	irregular	✓	✓	real-world
[120]	2025	binary	static	AA	area	• partial coverage • minimize costs • full connectivity	regular	✓	×	author-defined

Note: AA = Approximation Algorithm, GA = Genetic Algorithm, CG = Computational Geometry, MST = Minimum Spanning Tree, Node Loc. Cons. = Node Location Constraint.

Figure 22 illustrates the key environmental and spatial factors affecting SILs deployment in 3D terrains, including coverage holes due to insufficient placement, pest hotspot distribution, blind areas caused by occlusion, and physical constraints on region accessibility.

To achieve efficient deployment, the strategy must satisfy multiple objectives. On the one hand, the coverage of pest control zones should be maximized—especially in occluded or steep, e.g., sloped areas, to ensure there are no significant coverage holes. On the other hand, the number of deployed nodes and associated costs should be minimized to maintain economic feasibility. Furthermore, pest hotspot data should be considered, and Q-coverage strategies applied to deploy more SILs in high-risk areas, thereby enhancing trapping effectiveness. Additionally, deployment locations must meet constraints such as regional accessibility and stable solar exposure to ensure reliable power supply.

Future research on SIL deployment in 3D agricultural environments may extend in several directions. Firstly, incorporating terrain modeling techniques and DEM data can facilitate the development of 3D simulation platforms with realistic constraints. Secondly, integrating multi-source agricultural data, e.g., pest distribution, crop growth status,

and solar radiation, can enable data-driven, adaptive deployment strategies. Thirdly, dynamic deployment mechanisms using UAVs or mobile robotic platforms can support self-organizing and real-time adjustable deployment. Finally, a comprehensive deployment framework that simultaneously considers communication reliability, energy supply, and pest control effectiveness may contribute to more intelligent and systematic pest management in smart agriculture.

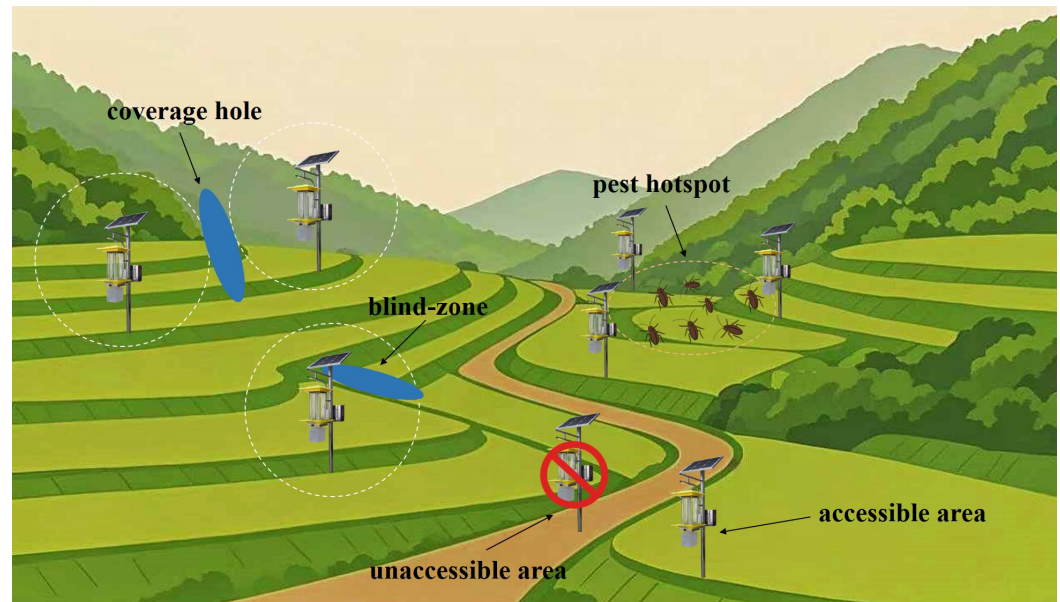


Figure 22. Problems in 3D SILs deployment.

7. Conclusions and Future Perspectives

7.1. Conclusions

In this paper, we presented a comprehensive survey of node deployment algorithms for 3D WSNs. We first summarized the fundamental design elements in 3D WSNs, including sensing models, blind spot detection on 3D surfaces, coverage and connectivity, sensor mobility, signal and protocol effects, and the construction of simulation maps. Subsequently, we categorized existing deployment algorithms into six types: classical algorithms, computational geometry algorithms, virtual force algorithms, evolutionary algorithms, swarm intelligence algorithms, and approximation algorithms. For each category, we reviewed the fundamental principles, representative techniques, and specific applications in 3D WSNs. A comparative analysis was conducted across these algorithm types, focusing on their optimization objectives, sensing models, coverage types, and deployment strategies. Through this analysis, we highlighted the strengths and limitations of each algorithm class, providing guidance for selecting appropriate methods based on various application scenarios. Finally, taking the deployment of solar insecticidal lamps in WSNs as a case study, we analyzed the key challenges posed by complex terrains and discussed deployment optimization strategies, as well as outlining directions for future research.

7.2. Future Challenges and Research Directions

Despite significant advancements in node deployment for 3D WSNs, several important research directions remain underexplored. Future studies are encouraged to focus on the following areas to enhance the effectiveness and applicability of deployment in 3D WSNs:

- (1) **Multi-objective optimization:** Real-world deployment problems often involve conflicting objectives, such as maximizing coverage and connectivity while minimizing energy

consumption and deployment costs. Developing adaptable solutions to effectively manage these trade-offs is crucial, particularly in complex and dynamic environments.

- (2) **Integration of real-world terrain data:** The use of idealized or synthetic terrains reduces the realism of simulation results. Incorporating accurate DEM and remote sensing data can significantly improve the accuracy of deployment simulations and support the development of more practical deployment strategies.
- (3) **Hybrid deployment architectures:** Combining static and mobile sensor nodes within hybrid centralized–distributed architectures offers enhanced adaptability and fault tolerance. Future work may explore efficient coordination mechanisms for hybrid deployments in complex and large-scale scenarios.
- (4) **Collaborative deployment with heterogeneous nodes:** Future systems are likely to increasingly integrate diverse sensor types with varying sensing ranges, energy capacities, and communication capabilities. Thus, an effective deployment strategy must accommodate multiple heterogeneous nodes (e.g., environmental sensors, directional cameras, and relay units), which is essential to improve the performance and robustness of WSNs.
- (5) **Multi-source information fusion:** Integrating diverse data sources, such as terrain models, environmental conditions, and infrastructure layouts, can enhance the adaptability and precision of deployment strategies. Future work should explore data-driven approaches to support decision-making for various applications in 3D WSNs.
- (6) **Integration with emerging technologies:** Future research may explore the integration of edge AI, UAV-assisted deployments, 5G/6G communication, and blockchain-based trust and security. Digital twins and cloud–edge frameworks could support real-time monitoring, predictive optimization, and adaptive decision-making, extending the applicability of deployment strategies in complex 3D WSNs.
- (7) **Security-aware deployment strategies:** Future work may also integrate security and trust management into 3D WSN deployment strategies, considering resilience against malicious attacks, intrusion detection, and blockchain-based trust frameworks during node placement and role assignment, thereby enhancing the reliability and robustness of deployments in adversarial environments.

Author Contributions: Author Contributions: Conceptualization, T.C., F.Y. and L.S.; Writing—original draft preparation, T.C.; Writing—review and editing, T.C., F.Y., L.S. and C.F.; Supervision, X.Y. and R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under Grant 62102170 and Grant 62072248, and in part by Research Start-Up Fund for Talent Introduction of Jiangsu Normal University under Grant 22XFRS033.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lanzolla, A.; Spadavecchia, M. Wireless sensor networks for environmental monitoring. *Sensors* **2021**, *21*, 1172. [[CrossRef](#)] [[PubMed](#)]
2. Deif, D.S.; Gadallah, Y. Classification of Wireless Sensor Networks Deployment Techniques. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 834–855. [[CrossRef](#)]
3. Mabrouki, J.; Azrou, M.; Dhiba, D.; Farhaoui, Y.; El Hajjaji, S. IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. *Big Data Min. Anal.* **2021**, *4*, 25–32. [[CrossRef](#)]
4. Ting, Y.T.; Chan, K.Y. Optimising performances of LoRa based IoT enabled wireless sensor network for smart agriculture. *J. Agric. Food Res.* **2024**, *16*, 101093. [[CrossRef](#)]
5. Sharma, H.; Haque, A.; Blaabjerg, F. Machine learning in wireless sensor networks for smart cities: A survey. *Electronics* **2021**, *10*, 1012. [[CrossRef](#)]

6. Bushnaq, O.M.; Chaaban, A.; Al-Naffouri, T.Y. The role of UAV-IoT networks in future wildfire detection. *IEEE Internet Things J.* **2021**, *8*, 16984–16999. [\[CrossRef\]](#)
7. Luo, J.; Chen, Y.; Wu, M.; Yang, Y. A survey of routing protocols for underwater wireless sensor networks. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 137–160. [\[CrossRef\]](#)
8. Shu, L.; Mukherjee, M.; Wu, X. Toxic gas boundary area detection in large-scale petrochemical plants with industrial wireless sensor networks. *IEEE Commun. Mag.* **2016**, *54*, 22–28. [\[CrossRef\]](#)
9. Jabeen, T.; Jabeen, I.; Ashraf, H.; Jhanjhi, N.; Yassine, A.; Hossain, M.S. An intelligent healthcare system using IoT in wireless sensor network. *Sensors* **2023**, *23*, 5055. [\[CrossRef\]](#)
10. Zhu, C.; Sheng, Z.; Leung, V.C.; Shu, L.; Yang, L.T. Toward offering more useful data reliably to mobile cloud from wireless sensor network. *IEEE Trans. Emerg. Top. Comput.* **2014**, *3*, 84–94. [\[CrossRef\]](#)
11. Xiang, S.; Yang, J. A novel adaptive deployment method for the single-target tracking of mobile wireless sensor networks. *Reliab. Eng. Syst. Saf.* **2023**, *234*, 109135. [\[CrossRef\]](#)
12. Olasupo, T.O.; Otero, C.E. A framework for optimizing the deployment of wireless sensor networks. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 1105–1118. [\[CrossRef\]](#)
13. Wu, J.; Wang, Y.; Liu, L. An Efficient Depth-adjustment Deployment Scheme for Underwater Wireless Sensor Networks. In Proceedings of the 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 7771–7776.
14. Iyer, S.; Rao, D.V. Genetic algorithm based optimization technique for underwater sensor network positioning and deployment. In Proceedings of the 2015 IEEE Underwater Technology, Chennai, India, 23–25 February 2015; pp. 1–6.
15. Latif, K.; Javaid, N.; Ahmad, A.; Khan, Z.A.; Alrajeh, N.; Khan, M.I. On energy hole and coverage hole avoidance in underwater wireless sensor networks. *IEEE Sens. J.* **2016**, *16*, 4431–4442. [\[CrossRef\]](#)
16. Han, G.; Zhang, C.; Shu, L.; Rodrigues, J.J. Impacts of deployment strategies on localization performance in underwater acoustic sensor networks. *IEEE Trans. Ind. Electron.* **2014**, *62*, 1725–1733. [\[CrossRef\]](#)
17. Brown, T.; Wang, Z.; Shan, T.; Wang, F.; Xue, J. On wireless video sensor network deployment for 3D indoor space coverage. In Proceedings of the SoutheastCon 2016, Norfolk, VA, USA, 30 March–3 April 2016; pp. 1–8.
18. Mnasri, S.; Nasri, N.; van den Bossche, A.; Val, T. A new multi-agent particle swarm algorithm based on birds accents for the 3D indoor deployment problem. *ISA Trans.* **2019**, *91*, 262–280. [\[CrossRef\]](#)
19. Afghantoloe, A.; Mostafavi, M.A. A purpose-oriented 3D Voronoi algorithm for deployment of a multi-type sensor network in complex 3D indoor environments in support of the mobility of people with motor disabilities. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 2519713. [\[CrossRef\]](#)
20. Boufares, N.; Khoufi, I.; Minet, P.; Saidane, L. 3D surface covering with virtual forces. In Proceedings of the Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), Hammamet, Tunisia, 4–6 November 2015.
21. Kim, K. Mountainous terrain coverage in mobile sensor networks. *IET Commun.* **2015**, *9*, 613–620. [\[CrossRef\]](#)
22. Li, X.; Ban, B.; Yang, Y.; Jin, M. Localization of networks on 3D terrain surfaces. *IEEE Trans. Mob. Comput.* **2020**, *21*, 1710–1722. [\[CrossRef\]](#)
23. Elhabyan, R.; Shi, W.; St-Hilaire, M. A full area coverage guaranteed, energy efficient network configuration strategy for 3D wireless sensor networks. In Proceedings of the 2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE), Quebec, QC, Canada, 13–16 May 2018; pp. 1–6.
24. Mostafa, B.; Hajraoui, A.; Chakkor, S. Three-Dimensional Application-Specific Protocol Architecture for Wireless Sensor Networks. *TELKOMNIKA Indones. J. Electr. Eng.* **2015**, *15*, 352.
25. Thalore, R.; Khurana, M.; Jha, M.K. Performance Comparison of 2D and 3D Zigbee Wireless Sensor Networks. In *Proceedings of International Conference on ICT for Sustainable Development*; Satapathy, S.C., Joshi, A., Modi, N., Pathak, N., Eds.; Springer: Singapore, 2016; pp. 215–222.
26. Zafer, M.; Senouci, M.R.; Aissani, M. On coverage of 3D terrains by wireless sensor networks. In Proceedings of the 2019 Federated Conference on Computer Science and Information Systems (FedCSIS), Leipzig, Germany, 1–4 September 2019; pp. 501–504.
27. Nasri, N.; Mnasri, S.; Val, T. 3D node deployment strategies prediction in wireless sensors network. *Int. J. Electron.* **2020**, *107*, 808–838. [\[CrossRef\]](#)
28. Zhang, X.; Zhang, B.; Chen, X.; Fang, Y. Coverage optimization of visual sensor networks for observing 3-D objects: Survey and comparison. *Int. J. Intell. Robot. Appl.* **2019**, *3*, 342–361. [\[CrossRef\]](#)
29. Han, F.; Liu, X.; Mohamed, I.I.; Ghazali, K.H.; Zhao, Y. A survey on deployment and coverage strategies in three-dimensional wireless sensor networks. In Proceedings of the 2019 8th International Conference on Software and Computer Applications, Penang, Malaysia, 19–21 February 2019; pp. 544–549.
30. Zhang, J.; Chen, X.; Zhang, W.H. A geodesic distance-based routing scheme for sensor networks with irregular terrain structure. *Wirel. Netw.* **2023**, *29*, 3207–3221. [\[CrossRef\]](#)

31. Mao, S.; Gou, P.; Zheng, C. Wireless Sensor Networks Deployment Strategy of 3D Surface Based on Delaunay Triangulation and Enhanced Butterfly Optimization Algorithm. In Proceedings of the 2024 9th International Conference on Communication, Image and Signal Processing (CCISP), Gold Coast, Australia, 13–15 November 2024; pp. 118–122.
32. Li, F.; Luo, J.; Wang, W.; He, Y. Autonomous Deployment for Load Balancing k-Surface Coverage in Sensor Networks. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 279–293. [[CrossRef](#)]
33. Hao, Z.; Qu, N.; Dang, X.; Hou, J. Node optimization coverage method under link model in passive monitoring system of three-dimensional wireless sensor network. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1–16. [[CrossRef](#)]
34. Gupta, H.P.; Venkatesh, T.; Rao, S.V.; Dutta, T.; Iyer, R.R. Analysis of Coverage Under Border Effects in Three-Dimensional Mobile Sensor Networks. *IEEE Trans. Mob. Comput.* **2017**, *16*, 2436–2449. [[CrossRef](#)]
35. Su, Y.; Guo, L.; Jin, Z.; Fu, X. A Voronoi-based optimized depth adjustment deployment scheme for underwater acoustic sensor networks. *IEEE Sens. J.* **2020**, *20*, 13849–13860. [[CrossRef](#)]
36. Bairagi, K.; Mitra, S.; Bhattacharya, U. Coverage aware scheduling strategies for 3D wireless video sensor nodes to enhance network lifetime. *IEEE Access* **2021**, *9*, 124176–124199. [[CrossRef](#)]
37. Xiao, F.; Yang, X.; Yang, M.; Sun, L.; Wang, R.; Yang, P. Surface coverage algorithm in directional sensor networks for three-dimensional complex terrains. *Tsinghua Sci. Technol.* **2016**, *21*, 397–406. [[CrossRef](#)]
38. Zhang, L.; Liu, T.T.; Wen, F.Q.; Hu, L.; Hei, C.; Wang, K. Differential evolution based regional coverage-enhancing algorithm for directional 3D wireless sensor networks. *IEEE Access* **2019**, *7*, 93690–93700. [[CrossRef](#)]
39. Dang, X.; Shao, C.; Hao, Z. Dynamic adjustment optimisation algorithm in 3D directional sensor networks based on spherical sector coverage models. *J. Sens.* **2019**, *2019*, 1018434. [[CrossRef](#)]
40. Cao, B.; Zhao, J.; Lv, Z.; Liu, X.; Kang, X.; Yang, S. Deployment optimization for 3D industrial wireless sensor networks based on particle swarm optimizers with distributed parallelism. *J. Netw. Comput. Appl.* **2018**, *103*, 225–238. [[CrossRef](#)]
41. Cao, B.; Zhao, J.; Yang, P.; Lv, Z.; Liu, X.; Min, G. 3-D multiobjective deployment of an industrial wireless sensor network for maritime applications utilizing a distributed parallel algorithm. *IEEE Trans. Ind. Inform.* **2018**, *14*, 5487–5495. [[CrossRef](#)]
42. Wang, Y.; Chen, Y. Coverage-All Targets Algorithm for 3D Wireless Multimedia Sensor Networks Based on the Gravitational Search Algorithm. *Autom. Control Comput. Sci.* **2019**, *53*, 429–440. [[CrossRef](#)]
43. Wang, Y.; Bi, X.; Teng, Z.; Wu, C. Coverage-all targets algorithm of directional sensor network for three-dimensional perception. *J. Jilin Univ. (Eng. Technol. Ed.)* **2015**, *45*, 1671–1679.
44. Wang, Z.; Xie, H. Wireless sensor network deployment of 3D surface based on enhanced grey wolf optimizer. *IEEE Access* **2020**, *8*, 57229–57251. [[CrossRef](#)]
45. Wang, Z.; Xiao, H.; Yang, S.; Wang, J.; Mahmoodi, S. Multistrategy integrated marine predator algorithm applied to 3D surface WSN coverage optimization. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9593103. [[CrossRef](#)]
46. Unaldi, N.; Temel, S.; Asari, V.K. Method for optimal sensor deployment on 3D terrains utilizing a steady state genetic algorithm with a guided walk mutation operator based on the wavelet transform. *Sensors* **2012**, *12*, 5116–5133. [[CrossRef](#)]
47. Saad, A.; Senouci, M.R.; Benyattou, O. Toward a realistic approach for the deployment of 3D wireless sensor networks. *IEEE Trans. Mob. Comput.* **2020**, *21*, 1508–1519. [[CrossRef](#)]
48. Zhang, L.G.; Fan, F.; Chu, S.C.; Garg, A.; Pan, J.S. Hybrid Strategy of Multiple Optimization Algorithms Applied to 3-D Terrain Node Coverage of Wireless Sensor Network. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6690824. [[CrossRef](#)]
49. Pan, J.S.; Chai, Q.W.; Chu, S.C.; Wu, N. 3-D terrain node coverage of wireless sensor network using enhanced black hole algorithm. *Sensors* **2020**, *20*, 2411. [[CrossRef](#)] [[PubMed](#)]
50. Wu, C.Q.; Wang, L. On efficient deployment of wireless sensors for coverage and connectivity in constrained 3D space. *Sensors* **2017**, *17*, 2304. [[CrossRef](#)]
51. Mnasri, S.; Nasri, N.; Alrashidi, M.; Val, T. 3D deployment problem in wireless sensor networks resolved by genetic and ant colony algorithms. In Proceedings of the 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 9–10 September 2020; pp. 1–5.
52. Mnasri, S.; Van Den Bossche, A.; Nasri, N.; Val, T. The 3D deployment multi-objective problem in mobile WSN: Optimizing coverage and localization. *Int. Res. J. Innov. Eng.-IRJIE* **2015**, *1*.
53. Afizudeen, S.; Pavithra, R. Grundy Number-based optimal sensor placement in 3D wireless sensor network. *IEEE Access* **2024**, *12*, 148502–148515. [[CrossRef](#)]
54. Balaji, S.; Priyanka; Anitha. Optimal deployment of sensors in 3D-terrain with q-coverage constraints. In Proceedings of the 2018 IEEE Sensors, New Delhi, India, 28–31 October 2018; pp. 1–4.
55. Cao, B.; Kang, X.; Zhao, J.; Yang, P.; Lv, Z.; Liu, X. Differential evolution-based 3-D directional wireless sensor network deployment optimization. *IEEE Internet Things J.* **2018**, *5*, 3594–3605. [[CrossRef](#)]
56. Gupta, H.P.; Rao, S.V.; Venkatesh, T. Analysis of stochastic coverage and connectivity in three-dimensional heterogeneous directional wireless sensor networks. *Pervasive Mob. Comput.* **2016**, *29*, 38–56. [[CrossRef](#)]

57. Arivudainambi, D.; Pavithra, R. Coverage and Connectivity-Based 3D Wireless Sensor Deployment Optimization. *Wirel. Pers. Commun.* **2020**, *112*, 1185–1204. [\[CrossRef\]](#)
58. Wu, C.Y.; Huang, Z.L.; Lin, G.; Ke, C.Q.; Lan, T.C. Coverage Maximization of WSNs in 3D Space Based on Hybrid Lion Swarm Optimization. *Wirel. Commun. Mob. Comput.* **2023**, *2023*, 8320637. [\[CrossRef\]](#)
59. Yao, Y.; Liao, H.; Liu, M.; Yang, X. Coverage optimization strategy for 3-D wireless sensor networks based on improved sparrow search algorithm. *IEEE Sens. J.* **2023**, *23*, 23721–23733. [\[CrossRef\]](#)
60. Hao, Z.; Qu, N.; Dang, X.; Hou, J. RSS-based coverage deployment method under probability model in 3D-WSN. *IEEE Access* **2019**, *7*, 183091–183104. [\[CrossRef\]](#)
61. Yang, X. A sion-free self-deployment of mobile robotic sensors for three-dimensional distributed blanket coverage control. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, China, 5–8 December 2017; pp. 80–85.
62. Hao, Z.; Xu, H.; Dang, X.; Qu, N. Method for Patching Three-Dimensional Surface Coverage Loopholes of Hybrid Nodes in Wireless Sensor Networks. *J. Sens.* **2020**, *2020*, 6492457. [\[CrossRef\]](#)
63. Tang, W.; Ma, X.; Wei, J.; Wang, Z. Measurement and analysis of near-ground propagation models under different terrains for wireless sensor networks. *Sensors* **2019**, *19*, 1901. [\[CrossRef\]](#)
64. Zhou, M.; Chen, C.; Chu, X. Impact of 3D antenna radiation pattern on heterogeneous cellular networks. *IEEE Access* **2022**, *10*, 120866–120879. [\[CrossRef\]](#)
65. Ruz-Nieto, A.; Egea-Lopez, E.; Molina-Garcia-Pardo, J.M.; Santa, J. A 3D simulation framework with ray-tracing propagation for LoRaWAN communication. *Internet Things* **2023**, *24*, 100964. [\[CrossRef\]](#)
66. Khalifeh, A.; Tanash, R.; AlQudah, M.; Al-Agtash, S. Enhancing energy efficiency of IEEE 802.15. 4-based industrial wireless sensor networks. *J. Ind. Inf. Integr.* **2023**, *33*, 100460.
67. Nayyar, A.; Singh, R. A comprehensive review of simulation tools for wireless sensor networks (WSNs). *J. Wirel. Netw. Commun.* **2015**, *5*, 19–47.
68. Manish, R.; Habib, A. In-situ calibration and trajectory enhancement of UAV LiDAR systems for mapping mechanized agricultural fields. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 7460–7474. [\[CrossRef\]](#)
69. Youn, W.; Ko, H.; Choi, H.; Choi, I.; Baek, J.H.; Myung, H. Collision-free autonomous navigation of a small UAV using low-cost sensors in GPS-denied environments. *Int. J. Control Autom. Syst.* **2021**, *19*, 953–968. [\[CrossRef\]](#)
70. Cao, B.; Zhao, J.; Yang, P.; Yang, P.; Liu, X.; Zhang, Y. 3-D deployment optimization for heterogeneous wireless directional sensor networks on smart city. *IEEE Trans. Ind. Inform.* **2018**, *15*, 1798–1808. [\[CrossRef\]](#)
71. Liu, N.; Chai, Q.W.; Liu, S.; Zheng, W.M. A novel compact particle swarm optimization for optimizing coverage of 3D in wireless sensor network. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 4600787. [\[CrossRef\]](#)
72. Kong, L.; Zhao, M.; Liu, X.Y.; Lu, J.; Liu, Y.; Wu, M.Y.; Shu, W. Surface coverage in sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *25*, 234–243. [\[CrossRef\]](#)
73. Ribeiro, M.G.; Neves, L.A.; Pinto, A.; Nascimento, M.Z.d.; Zafalon, G.F.D.; Valêncio, C. Surface coverage in wireless sensor networks based on Delaunay tetrahedralization. *J. Phys. Conf. Ser.* **2015**, *574*, 012083. [\[CrossRef\]](#)
74. Fu, W.; Yang, Y.; Hong, G.; Hou, J. WSN deployment strategy for real 3D terrain coverage based on greedy algorithm with DEM probability coverage model. *Electronics* **2021**, *10*, 2028. [\[CrossRef\]](#)
75. Zafer, M.; Senouci, M.R.; Aissani, M. Terrain partitioning based approach for realistic deployment of wireless sensor networks. In Proceedings of the Computational Intelligence and Its Applications: 6th IFIP TC 5 International Conference, Oran, Algeria, 8–10 May 2018; pp. 423–435.
76. Veenstra, K.; Obraczka, K. Guiding sensor-node deployment over 2.5D terrain. In Proceedings of the IEEE International Conference on Communications, London, UK, 8–12 June 2015; pp. 6719–6725.
77. Lakshmi, S.E.; Yarrakula, K. Review and critical analysis on digital elevation models. *Geofizika* **2018**, *35*, 129–157. [\[CrossRef\]](#)
78. Cao, B.; Zhao, J.; Lv, Z.; Liu, X. 3D terrain multiobjective deployment optimization of heterogeneous directional sensor networks in security monitoring. *IEEE Trans. Big Data* **2017**, *5*, 495–505. [\[CrossRef\]](#)
79. Boufares, N.; Saied, Y.B.; Saidane, L.A. Improved Distributed Virtual Forces Algorithm for 3D Terrains Coverage in Mobile Wireless Sensor Networks. In Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications, Aqaba, Jordan, 28 October–1 November 2018; pp. 1–8.
80. Ammari, H.M. Connected coverage in three-dimensional wireless sensor networks using convex polyhedral space-fillers. In Proceedings of the 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), Ottawa, ON, Canada, 5–7 June 2017; pp. 183–190.
81. Arivudainambi, D.; Pavithra, R. Vertex coloring approach for Q-coverage problem in wireless sensor network. *J. Intell. Fuzzy Syst.* **2021**, *40*, 8683–8695. [\[CrossRef\]](#)
82. Katti, A.; Lobiyal, D. Node deployment strategies and coverage prediction in 3D wireless sensor network with scheduling. *Adv. Comput. Sci. Technol.* **2017**, *10*, 2243–2255.

83. Okabe, A.; Satoh, T.; Furuta, T.; Suzuki, A.; Okano, K. Generalized network Voronoi diagrams: Concepts, computational methods, and applications. *Int. J. Geogr. Inf. Sci.* **2008**, *22*, 965–994. [\[CrossRef\]](#)
84. Anand, N.; Ranjan, R.; Rai, B.S.; Varma, S. A novel computational geometry-based node deployment scheme in 3D wireless sensor network. *Int. J. Sens. Netw.* **2017**, *25*, 135–145. [\[CrossRef\]](#)
85. Yu, X.; Huang, W.; Lan, J.; Qian, X. A novel virtual force approach for node deployment in wireless sensor network. In Proceedings of the 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems, Hangzhou, China, 16–18 May 2012; pp. 359–363.
86. Miao, C.; Dai, G.; Zhao, X.; Tang, Z.; Chen, Q. 3D self-deployment algorithm in mobile wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 721921. [\[CrossRef\]](#)
87. Mirjalili, S.; Mirjalili, S. Genetic algorithm. In *Evolutionary Algorithms and Neural Networks: Theory and Applications*; Springer: Cham, Switzerland, 2019; pp. 43–55.
88. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2010**, *15*, 4–31. [\[CrossRef\]](#)
89. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [\[CrossRef\]](#)
90. Zhou, J.; Qi, G.; Liu, C. A chaotic parallel artificial fish swarm algorithm for water quality monitoring sensor networks 3D coverage optimization. *J. Sens.* **2021**, *2021*, 5529527. [\[CrossRef\]](#)
91. Mohar, S.S.; Goyal, S.; Kaur, R. Optimum deployment of sensor nodes in wireless sensor network using hybrid fruit fly optimization algorithm and bat optimization algorithm for 3D Environment. *Peer-to-Peer Netw. Appl.* **2022**, *15*, 2694–2718. [\[CrossRef\]](#)
92. Ru, J.; Jia, Z.; Yang, Y.; Yu, X.; Wu, C.; Xu, M. A 3D coverage algorithm based on complex surfaces for UAVs in wireless multimedia sensor networks. *Sensors* **2019**, *19*, 1902. [\[CrossRef\]](#) [\[PubMed\]](#)
93. Liu, Y.; Chong, E.K.; Pezeshki, A.; Zhang, Z. Submodular optimization problems and greedy strategies: A survey. *Discret. Event Dyn. Syst.* **2020**, *30*, 381–412. [\[CrossRef\]](#)
94. Liu, B.H.; Tang, Y.J.; Yu, C.W.; Tsai, M.J. Greedy algorithms for actor redeployment in wireless sensor-actor networks. *Wirel. Netw.* **2015**, *21*, 431–442. [\[CrossRef\]](#)
95. Priyadarshi, R.; Gupta, B. Area Coverage Optimization in Three-Dimensional Wireless Sensor Network. *Wirel. Pers. Commun.* **2021**, *117*, 843–865. [\[CrossRef\]](#)
96. Saikia, M.; Hussain, M.A. Wireless sensor node deployment strategy for hilly terrains—a surface approximation based approach. *IET Wirel. Sens. Syst.* **2019**, *9*, 284–294. [\[CrossRef\]](#)
97. Gou, P.; Guo, B.; Guo, M.; Mao, S. VKECE-3D: Energy-efficient coverage enhancement in three-dimensional heterogeneous wireless sensor networks based on 3D-voronoi and K-means algorithm. *Sensors* **2023**, *23*, 573. [\[CrossRef\]](#)
98. Tan, L.; Shi, J.; Tang, X.; Lian, X.; Wang, H. 3dDABA: An Algorithm Covering a Three-dimensional WSN Area. *J. Internet Technol.* **2020**, *21*, 1949–1956.
99. Boufares, N.; Minet, P.; Khoufi, I.; Saidane, L. Covering a 3D flat surface with autonomous and mobile wireless sensor nodes. In Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 1628–1633.
100. Dang, X.; Shao, C.; Hao, Z. Target Detection Coverage Algorithm Based on 3D-Voronoi Partition for Three-Dimensional Wireless Sensor Networks. *Mob. Inf. Syst.* **2019**, *2019*, 7542324. [\[CrossRef\]](#)
101. Zhang, M.; Yang, J.; Qin, T. An adaptive three-dimensional improved virtual force coverage algorithm for nodes in WSN. *Axioms* **2022**, *11*, 199. [\[CrossRef\]](#)
102. Tsang, Y.P.; Choy, K.L.; Wu, C.H.; Ho, G.T.S. Multi-objective mapping method for 3D environmental sensor network deployment. *IEEE Commun. Lett.* **2019**, *23*, 1231–1235. [\[CrossRef\]](#)
103. Mnasri, S.; Nasri, N.; van den Bossche, A.; Val, T. Improved many-objective optimization algorithms for the 3D indoor deployment problem. *Arab. J. Sci. Eng.* **2019**, *44*, 3883–3904. [\[CrossRef\]](#)
104. Rehman, E.; Sher, M.; Naqvi, S.H.A.; Badar Khan, K.; Ullah, K. Energy efficient secure trust based clustering algorithm for mobile wireless sensor network. *J. Comput. Netw. Commun.* **2017**, *2017*, 1630673. [\[CrossRef\]](#)
105. Zhang, Y.; Wang, M.; Liang, J.; Zhang, H.; Chen, W.; Jiang, S. Coverage enhancing of 3D underwater sensor networks based on improved fruit fly optimization algorithm. *Soft Comput.* **2017**, *21*, 6019–6029. [\[CrossRef\]](#)
106. Arivudainambi, D.; Balaji, S.; Poorani, T. Sensor deployment for target coverage in underwater wireless sensor network. In Proceedings of the 2017 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), Paris, France, 28–30 November 2017; pp. 1–6.
107. Ren, J.; Li, Y. Energy-Efficient Coverage Enhancement Strategy for WSNs Based on a Competitive Learning Optimizer. *IEEE Internet Things J.* **2025**, *12*, 14548–14558. [\[CrossRef\]](#)
108. Du, Y. Method for the optimal sensor deployment of WSNs in 3D terrain based on the DPSOVF algorithm. *IEEE Access* **2020**, *8*, 140806–140821. [\[CrossRef\]](#)

109. Li, J.; Li, G.C.; Chu, S.C.; Gao, M.; Pan, J.S. Modified parallel tunicate swarm algorithm and application in 3D WSNs coverage optimization. *J. Internet Technol.* **2022**, *23*, 227–244.
110. Yang, F.; Shu, L.; Wang, X. AnaMap: A methodology of simulation and visualization for actual farmland topography. In Proceedings of the 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), Palma de Mallorca, Spain, 21–23 July 2021; pp. 1–7.
111. Yang, F.; Shu, L.; Huang, K.; Li, K.; Han, G.; Liu, Y. A partition-based node deployment strategy in solar insecticidal lamps Internet of Things. *IEEE Internet Things J.* **2020**, *7*, 11223–11237. [[CrossRef](#)]
112. Yang, F.; Shu, L.; Yang, Y.; Liu, Y.; Gordon, T. Improved coverage and connectivity via weighted node deployment in solar insecticidal lamp Internet of Things. *IEEE Internet Things J.* **2021**, *8*, 10170–10186. [[CrossRef](#)]
113. Yang, F.; Shu, L.; Yang, Y.; Han, G.; Pearson, S.; Li, K. Optimal deployment of solar insecticidal lamps over constrained locations in mixed-crop farmlands. *IEEE Internet Things J.* **2021**, *8*, 13095–13114. [[CrossRef](#)]
114. Yang, F.; Shu, L.; Su, Q.; Han, G. Optimal deployment of IoT-based solar insecticide lamps under coverage and maintenance cost considerations. In Proceedings of the 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), Perth, Australia, 25–28 July 2022; pp. 251–256.
115. Si, P.; Fu, Z.; Shu, L.; Yang, Y.; Huang, K.; Liu, Y. Target-barrier coverage improvement in an insecticidal lamps internet of UAVs. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4373–4382. [[CrossRef](#)]
116. Yang, F.; Shu, L.; Duan, N.; Yang, X.; Hancke, G.P. Complete Area ϵ -Probability Coverage in Solar Insecticidal Lamps Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 22764–22774. [[CrossRef](#)]
117. Yang, F.; Shu, L. A Trajectory-Inspired Node Deployment Strategy in Solar Insecticidal Lamps Internet of Things Under Coverage and Maintenance Cost Considerations. *IEEE Trans. AgriFood Electron.* **2024**, *2*, 28–42. [[CrossRef](#)]
118. Zhang, Y.; Si, P.; Shu, L. SIL-IoVTs: A Bi-Level Optimization Approach for Pest Control and Intruder Monitoring in Smart Agriculture. In Proceedings of the 2025 IEEE International Conference on Industrial Technology (ICIT), Wuhan, China, 26–28 March 2025; pp. 1–5.
119. Si, P.; Zhang, Y.; Zhang, H.; Su, Q.; Jing, X.; Shu, L. Optimal Deployment of Solar Insecticidal Lamps with Cameras in Smart Agriculture. *IEEE Trans. Green Commun. Netw.* **2025**. [[CrossRef](#)]
120. Yang, F.; Tian, X.; Zhang, Z.; Shu, L.; Jing, X. A Node Deployment Strategy in Solar Insecticidal Lamps Internet of Things with Respect to Partial Coverage and Energy Harvesting Requirements. *IEEE Trans. Sustain. Comput.* **2025**. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.